

USING THE GENERALIZED INTERPOLATION MATERIAL POINT METHOD  
FOR FLUID-SOLID INTERACTIONS INDUCED BY SURFACE TENSION

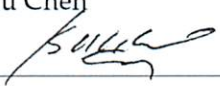
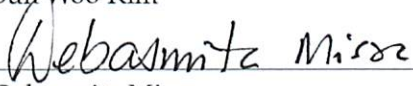
By

Liangbiao Chen

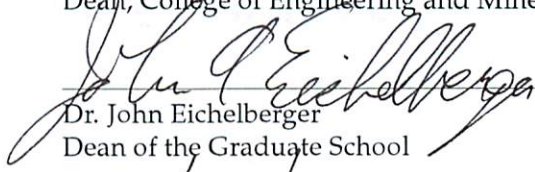
RECOMMENDED:



Dr. Cheng-fu Chen

  
Dr. Sun Woo Kim  
Dr. Debasmita Misra  
Dr. Jonah Lee  
Advisory Committee Chair  
Dr. Rorik Peterson  
Chair, Department of Mechanical Engineering

APPROVED:

  
Dr. Doug Goering  
Dean, College of Engineering and Mines  
Dr. John Eichelberger  
Dean of the Graduate School

11/28/13  
Date



USING THE GENERALIZED INTERPOLATION MATERIAL POINT METHOD  
FOR FLUID-SOLID INTERACTIONS INDUCED BY SURFACE TENSION

A  
THESIS

Presented to the Faculty  
of the University of Alaska Fairbanks  
in Partial Fulfillment of the Requirements  
for the Degree of

DOCTOR OF PHILOSOPHY

By  
Liangbiao Chen, B.S., M.Sc.

Fairbanks, Alaska

December 2013

## Abstract

This thesis is devoted to the development of new, Generalized Interpolation Material Point Method (GIMP)-based algorithms for handling surface tension and contact (wetting) in fluid-solid interaction (FSI) problems at small scales. In these problems, surface tension becomes so dominant that its influence on both fluids and solids must be considered. Since analytical solutions for most engineering problems are usually unavailable, numerical methods are needed to describe and predict complicated time-dependent states in the solid and fluid involved due to surface tension effects. Traditional computational methods for handling fluid-solid interactions may not be effective due to their weakness in solving large-deformation problems and the complicated coupling of two different types of computational frameworks: one for solid, and the other for fluid. On the contrary, GIMP, a mesh-free algorithm for solid mechanics problems, is numerically effective in handling problems involving large deformations and fracture. Here we extend the capability of GIMP to handle fluid dynamics problems with surface tension, and to develop a new contact algorithm to deal with the wetting boundary conditions that include the modeling of contact angle and slip near the triple points where the three phases – fluid, solid, and vapor – meet. The error of the new GIMP algorithm for FSI problems at small scales, as verified by various benchmark problems, generally falls within the 5% range. In this thesis, we have successfully extended the capability of GIMP for handling FSI problems under surface tension in a one-solver numerical framework, a unique and innovative approach.





## Table of Contents

	Page
<b>Signature Page</b> . . . . .	<b>i</b>
<b>Title Page</b> . . . . .	<b>iii</b>
<b>Abstract</b> . . . . .	<b>v</b>
<b>Table of Contents</b> . . . . .	<b>vii</b>
<b>List of Figures</b> . . . . .	<b>ix</b>
<b>List of Tables</b> . . . . .	<b>xv</b>
<b>Acknowledgements</b> . . . . .	<b>xvii</b>
<b>Chapter 1 Introduction</b> . . . . .	<b>1</b>
1.1 Background . . . . .	1
1.2 Objectives . . . . .	2
1.3 References . . . . .	4
<b>Chapter 2 Using the Generalized Interpolation Material Point Method for Fluid Dynamics at Low Reynolds Numbers</b> . . . . .	<b>7</b>
2.1 Abstract . . . . .	7
2.2 Introduction . . . . .	7
2.3 MPM for fluid dynamics . . . . .	8
2.3.1 MPM formulations . . . . .	8
2.3.2 Constitutive model for fluid . . . . .	11
2.3.3 Pressure stabilization . . . . .	12
2.4 Numerical examples and validations . . . . .	12
2.4.1 GIMP-CFD . . . . .	13
2.4.2 Alternative interpolation function – B-spline . . . . .	26
2.5 Discussions and conclusions . . . . .	27
2.6 References . . . . .	35
<b>Chapter 3 On the Modeling of Surface Tension and its Applications by the Generalized Interpolation Material Point Method</b> . . . . .	<b>39</b>
3.1 Abstract . . . . .	39
3.2 Introduction . . . . .	39
3.3 Review of the GIMP Method . . . . .	40
3.4 Modeling of Surface Tension . . . . .	44
3.4.1 Review of the CSF method . . . . .	44

	Page
3.4.2 Formulation of surface tension in GIMP . . . . .	45
3.4.3 Treatment of contact angle as a boundary condition . . . . .	47
3.4.4 Implementation of GIMP-CSF . . . . .	48
3.5 Numerical Examples . . . . .	50
3.5.1 2D equilibrium droplet . . . . .	50
3.5.2 Evolution of non-equilibrium drops in 2D and 3D . . . . .	55
3.5.3 Wall adhesion and capillary rise . . . . .	57
3.6 Discussions and Conclusion . . . . .	59
3.7 References . . . . .	62
<b>Chapter 4 Using the Generalized Interpolation Material Point Method for Fluid-Solid Interactions Induced by Surface Tension . . . . .</b>	<b>65</b>
4.1 Introduction . . . . .	65
4.2 Review of surface tension and contact angle . . . . .	66
4.3 Basic equations of MPM . . . . .	67
4.4 Surface tension model in GIMP-FSI . . . . .	71
4.4.1 The CSF model for fluid . . . . .	71
4.4.2 Treatment of contact angle . . . . .	73
4.4.3 Surface tension acting on solid . . . . .	74
4.5 Contact algorithm with partial slip . . . . .	75
4.6 Numerical examples . . . . .	77
4.6.1 Half-plane capillary rise . . . . .	78
4.6.2 Modeling of liquid bridges . . . . .	81
4.6.3 Bending of micro-cantilever plate . . . . .	90
4.7 Discussion and conclusions . . . . .	93
4.8 References . . . . .	96
<b>Chapter 5 Conclusions . . . . .</b>	<b>101</b>

## List of Figures

	Page
2.1 (a) Problem setup for hydrostatic pressure simulation (left), discretization of the domain (fluid particles and background mesh) with boundary conditions (right); (b) comparison of pressure vs. depth (measured from the bottom of the tank) at equilibrium state between theory and numerical results at the particles using the reduced bulk modulus $K_r=1.54 \times 10^4$ Pa. . . .	14
2.2 (a) Problem setup for Rayleigh's problem (left), discretization of the domain (fluid particles and background mesh) with boundary conditions (right); (b) calculated normalized velocity profile versus theoretical profile. . . . .	16
2.3 Simulation of plane Poiseuille flow by GIMP. . . . .	18
2.4 GIMP simulation of transient 3D Poiseuille flow in a pipe. . . . .	22
2.5 GIMP simulation of the lid-driven cavity problem for $Re = 100$ . . . . .	29
2.6 Setup of the dam-break problem for OpenFOAM and GIMP. . . . .	30
2.7 Simulation of dam-break using OpenFOAM and GIMP. . . . .	31
2.8 B-spline results for: (a) hydrostatic pressure with comparison to GIMP and theory; (b) Rayleigh's problem with comparison to theory. . . . .	32
2.9 B-spline results for 2D and 3D transient Poiseuille flows with comparison to theoretical results. . . . .	33
2.10 B-spline results without pressure stabilization for: (a) the lid-driven cavity problem as a function of PPC when compared with test data; (b) the dam-break problem when compared with results from GIMP with pressure stabilization. . . . .	34

- 3.1 Illustration of the procedure of GIMP. At the beginning of computation, the disk  $\Omega$  is discretized into a collection of particles (i.e., material points) which are distributed over a background mesh. Each particle carries the local material properties and state variables of the disk: stress  $\sigma_p$ , mass  $m_p$ , volume  $V_p$ , deformation gradient  $\mathbf{F}_p$ , position  $\mathbf{x}_p$ , and velocity  $\mathbf{v}_p$ . An interpolation function is then selected to map, in each time increment, the information from the particles to the grid points such that acceleration at each grid point can be solved via the equations of motion  $\mathbf{F}_i = m_i \mathbf{a}_i$ . Information at the grid points is then mapped back to the particles to complete the calculation of one time increment. . . . . 41
- 3.2 The CSF method for fluids with different colors ( $c_1$  and  $c_2$ ). Over the computational grid, contours of the continuous color function  $c(\mathbf{x})$  are illustrated showing also a transition zone of width  $h$ . Surface tension force,  $\mathbf{F}^{sa}$ , is reformulated as a volume force  $\mathbf{F}^{sv}$  within the transition zone. . . . . 44
- 3.3 Contact angle  $\theta_{eq}$  as a boundary condition.  $\mathbf{n}_{eq}$  is the unit vector normal to the tangent along  $\theta_{eq}$ ;  $\mathbf{n}_t$  is a unit vector tangent to the wall;  $\mathbf{n}_w$  is a unit vector normal to the wall; and  $\mathbf{n}_i$  is the unit normal vector along the liquid-wall contact line. During simulation, the profile of the liquid in equilibrium can be determined when  $\mathbf{n}_i$  is aligned with  $\mathbf{n}_{eq}$ . . . . . 47
- 3.4 A flow chart for GIMP-CSF.  $\nabla m_i$ —gradient of grid mass;  $\tilde{m}_i$ —smoothed grid mass;  $\mathbf{n}_i$ —grid surface normal;  $\mathbf{n}_{eq}$ —surface normal based on contact angle;  $\kappa_i$ —grid curvature;  $\mathbf{F}_i^{sv}$ —grid surface tension (volume) force;  $\mathbf{a}_i^{sur}$ —grid surface tension (acceleration) force. . . . . 48
- 3.5 Example set-up of 2D equilibrium drop with GIMP using Uintah. . . . . 51
- 3.6 Contours of the grid mass and the gradients at  $t=0$  s. . . . . 51
- 3.7 Curvatures ( $t=0$ ) with different smoothed grid mass: (a) zero iteration of moving averaging; (b) 1 iteration of moving averaging ; (c) 2 iterations of moving averaging; (d) 4 iterations of moving averaging. Theoretical curvature is  $1 \text{ cm}^{-1}$ . . . . . 52
- 3.8 Surface tension,  $\mathbf{a}^{sur}$ , at  $t=0$  with different curvatures from (a) one iteration of moving averaging; (b) 4 iterations of moving averaging. . . . . 53

3.9	Pressure at $t=2.5$ s from different smoothing schemes: (a) one iteration of moving averaging; (b) 4 iterations of moving averaging. Theoretical pressure drop is $2.4 \text{ dynes/cm}^2$ . . . . .	53
3.10	A 2D $1 \times 1$ cm square drop evolves into a circular shape under surface tension using two iterations of moving average smoothing. The arrows represent the surface tension force. . . . .	55
3.11	Distribution of the calculated curvature (a), and pressure (b) after a 2D non-equilibrium drop becomes stable at time = $0.25$ s. The average pressure is $4.09$ (or $96.1\%$ of the theoretical value), with an rms error $L_2 = 0.044$ . . . . .	56
3.12	Evolution of a 3D cubic drop into a sphere under surface tension using two iterations of moving average smoothing. . . . .	57
3.13	Distribution of the calculated curvature (a), and pressure (b) on a plane through the origin of a 3D drop. The average pressure is $8.05$ (or $104.1\%$ of the theoretical value) with an rms error $L_2 = 0.0451$ . . . . .	57
3.14	Schematic of half-plane liquid meniscus $y(x)$ under gravitational pull. The liquid is at rest in contact with a hydrophilic wall with a given contact angle $\theta_{eq}$ . $H$ is the capillary rise to be simulated. . . . .	58
3.15	Set-up for the simulation of the capillary rise. The contact angle $\theta_{eq}$ is prescribed at the wall boundary; the surface profile will evolve toward its equilibrium state under surface tension. . . . .	59
3.16	Evolution of the surface profile due to surface tension. Arrows indicate the surface tension force which are in equilibrium with gravitational pull at $t=0.1$ s ( $\theta_{eq} = 30^\circ$ ). . . . .	60
3.17	Comparison of the calculated capillary rise at $t=0.1$ s to theoretical solutions for different contact angles ( $30^\circ$ and $60^\circ$ ). . . . .	60
4.1	Schematic of surface tension and contact angle for a liquid-air-solid system. Fluid is subject to capillary pressure $\Delta P$ , while solid is subject to capillary pressure and the attraction force equal to $\sigma_{la} \sin \theta_{eq}$ . . . . .	67
4.2	Example of GIMP-FSI. $\mathbf{x}_p$ , particle position; $\mathbf{v}_p$ -particle velocity; $\sigma_p$ -particle stress; $V_p$ -particle volume; $\mathbf{f}_i^{sur,(f)}$ -surface tension force on grid nodes acting around fluid interface; $\mathbf{f}_i^{sur,(s)}$ -surface tension force acting on the solid near the triple points. . . . .	68

	Page
4.3 a) Curvature $\kappa$ calculated by divergence of $\tilde{\mathbf{n}}^f$ from the smoothed fluid mass is problematic at the triple points; (b) curvature $\kappa$ computed by divergence of the smoothed $\mathbf{n}^f = \mathbf{G}_{ip} m_p^f$ with a smoothing kernel $\psi$ . $\mathbf{G}_{ip} m_p^f$ is not calculated where solid mass ( $m_i^s$ ) is non-zero to avoid the sudden change of interface at the triple points. . . . .	72
4.4 Schematic of triple point identification to apply surface tension force $\mathbf{f}^{sur,s}$ on solid by a simple criterion. $\mathbf{n}^s$ - solid normal; . . . . .	75
4.5 Slip of fluid near the triple points. The slip zone involves the nodes where $\mathbf{n}^s$ and $\tilde{\mathbf{n}}_i^{slip}$ overlap. . . . .	76
4.6 (a) Half-plane capillary rise problem; (b) example setup in GIMP-FSI. . . . .	78
4.7 Kernel smoothing ( $\psi$ ) of fluid normal $\mathbf{n}^f$ for surface tension force $\mathbf{a}_i^{sur,(f)}$ and sliding zone. . . . .	79
4.8 Evolution of capillary rise. Partial slip can be seen near the triple points. Arrows are surface tension forces. . . . .	80
4.9 Comparison of capillary rise between GIMP-FSI and theory. Various PPC numbers and contact angles are presented. . . . .	81
4.10 (a)Schematic of liquid bridge between two parallel plates; (b) numerical setup in GIMP-FSI for half of the domain. . . . .	82
4.11 (a-d) Evolution of liquid profile with time; (e) pressure contour in fluid at equilibrium; (f) normal stress $S_{11}$ in the $x$ direction at equilibrium. Arrows are surface tension forces. . . . .	84
4.12 Calculated capillary pressure $\Delta P$ versus theory with various PPC numbers for liquid bridges between two parallel plates. . . . .	85
4.13 Calculated normal stress $S_{11}$ in the $x$ direction in two parallel plates. The total capillary force $F^{cap}$ caused by the liquid bridge is equal to the area under the curve. . . . .	85
4.14 Simulation of liquid bridge between two solid cylinders with a separation distance $2d = 1$ cm and a height $h = 1$ cm. . . . .	86
4.15 (a-d): Evolution of a liquid bridge between two cylinders of equal radii; e): pressure in the fluid; f): normal stress in the solid. . . . .	88

4.16	Calculated capillary pressure $\Delta P$ versus theory as a function of PPC for liquid bridges between two cylinders. . . . .	89
4.17	Calculated normal stress $S_{11}$ in the solid as a function of PPC. . . . .	89
4.18	Schematic of cantilever-beam bent by a droplet. . . . .	90
4.19	Numerical setup of the cantilever-beam problem in GIMP-CSF. . . . .	91
4.20	Evolution of parameters for the cantilever beam problem. . . . .	93
4.21	Cantilever beam problem with different numbers of $\lambda^{-1}c_0$ . . . . .	94
4.22	Calculated beam deflections versus theory for different numbers of $\lambda^{-1}c_0$ . .	94





## List of Tables

	Page
2.1 Effects of mesh size and PPC for the hydrostatic problem expressed in $L_\infty$ and $L_2$ using the reduced bulk modulus $K_r=1.54 \times 10^4$ Pa. . . . .	20
2.2 Effects of mesh size and PPC on Rayleigh's problems. For each case, minimum and maximum of $L_\infty$ and $L_2$ are given. . . . .	20
2.3 Effects of mesh size and PPC on plane Poiseuille flow using GIMP. . . . .	20
2.4 Effects of mesh size and PPC on 3D Poiseuille flow using GIMP. . . . .	23
2.5 Summary of computation time using GIMP-CFD. . . . .	26
3.1 Fluid properties used in the numerical examples . . . . .	50
3.2 Comparison of results using GIMP-CSF to those using the arbitrary Lagrangian-Eulerian (ALE) method for simulating a 2D equilibrium droplet. ( $\Delta h$ is the mesh size; PPC–particles per cell.) . . . . .	54
3.3 Summary of computational time. $\Delta h$ is mesh size, $t_p$ is physical time, and $t_c$ is computational time. . . . .	61
4.1 Capillary pressures ( $\Delta P$ ) and forces ( $F^{cap}$ ) by GIMP-FSI and theory for liquid bridge between two parallel plates. Unit “dyne” for force and “cm” for length. . . . .	86
4.2 Computed capillary pressure and force versus theory for a given volume of liquid between two cylinders of equal radii. Unit “dyne” for force and “cm” for length. . . . .	89
4.3 Parameters used in GIMP-FSI for the modeling of the cantilever beam problem for different values of $c_0 \lambda^{-1}$ . . . . .	92
4.4 Example of CPU time for different cases. . . . .	95



## Acknowledgements

I came to Fairbanks from China in Fall 2008 to study Geotechnical Engineering as a graduate student. Two years later, right after I obtained my M.S. degree in Geotechnical Engineering in 2010, I found myself also strongly interested in computational mechanics. Driven by this strong feeling, I switched to Mechanical Engineering in Fall 2010, and from that time began a wonderful journey of my PhD study that has been proven fruitful and rewarding.

I would like to first express my sincere thanks to Dr. Jonah Lee and Dr. Cheng-fu Chen. Both have provided me with invaluable support and help during my studies. Dr. Lee introduced me to a fascinating field of which I never feel tired. Dr. Chen provided no less help and guidance that allow me to achieve my academic goal. I am so fortunate that I can have them as my advisors.

I would also like to thank the other committee members, Dr. Debasmita Misra and Dr. Sun Woo Kim, for their willingness to join me in the journey with their professional expertise and precious time.

So many people have provided me with love and friendship during my studies in Fairbanks. Jie, my dear girl friend, has been as supportive as she could, in Alaska then or in Texas now. I owe her a big promise which I wish will be fulfilled soon. Other helpful hands come from Frances Bedel, our office manager, and my fellow graduate students. I apologize in advance if I fail to mention everyone: Jagannadha Satti, Ravikanth Vajjha, Vamshi Avadhanula, Vamsi Mokkalapati, Feng Xiao, Zhili Quan, Lin Li, and Zhipeng Dai. For my other personal friends, those who play table tennis with me, and those who treat me like a brother, I am so thankful to them for caring about me.

Finally, this work is dedicated to my dearest mother and father. I wish they will be proud of me at this point. I know they will, as they always are.

This work was funded by the Department of Mechanical Engineering in University of Alaska Fairbanks, and was supported in part by the Institute of Northern Engineering, and a grant of HPC resources from the Arctic Region Supercomputing Center. I am truly grateful for their support.



## Chapter 1

### Introduction

#### 1.1 Background

Fluid-structure interactions (FSI) can occur at small scales in unsaturated soil or porous rock, as well as in micro/nano manipulation processes. For these situations, surface tension and contact angles become crucial in determining the interactions between fluids and solids. The studies of fluid-solid interactions induced by surface tension can be applied to many industrial processes, including oil recovery and chemical assay by lab-on-a-chip devices [Kapur and Gaskell, 2007].

Theory of surface tension and contact angle was first described by Young [1805], who developed the famous Young-Laplace equation for determining the liquid interface. Young [1805] also proposed the Young's equations to include contact angle as a boundary condition at the triple points of three phases (e.g., solid, fluid, and air). The Young-Laplace equation and Young's equation are the two fundamental equations for including the effects of surface tension and contact angles in FSI problems.

When it comes to engineering problems, which usually involve a complex topology of fluid interfaces and contact conditions at fluid-solid interfaces, theoretical analysis can be very difficult. Thus, numerical approaches are often sought to solve these problems. To facilitate computational modeling of complicated immiscible interfaces in the presence of contact angle, Brackbill et al. [1992] proposed the Continuous Surface Force (CSF) method that has been widely adopted by most computational fluid dynamics codes. For solving common FSI problems without surface tension, meshed methods have been developed, such as the immersed boundary method [Xu and Wang, 2008, Gil et al., 2013], the Lattice Boltzmann method [Shin et al., 2013], arbitrary Lagrangian-Eulerian (ALE) method, and level set method [Basting and Weismann, 2013]. Generally, these methods are effective for small deformation problems. For large deformation problems, they often suffer from mesh entangling issues. Furthermore, these methods adopt different solvers for the solid and fluid phases requiring oftentimes complicated coupling equations to bridge the solid and fluid phases.

There is a current trend to use particle method or mesh-free method to solve problems where traditional meshed methods fail. In general, particle methods can minimize mesh entangling issues and can deal with complicated contact problems involving multiple materials and bodies. One of the popular particle methods is the smoothed-particle

hydrodynamics (SPH) method. SPH has been applied to FSI problems [Lahooti et al., 2011] and surface tension problems involving contact angles [Das and Das, 2010]. Compared to SPH, the material point method (MPM) is an emerging method developed by Sulsky et al. [1994, 1995] by extending the hydrodynamics fluid implicit particle (FLIP) method [Brackbill and Ruppel, 1986] to solid mechanics. Although MPM is devised to solve difficult solid problems, it can also handle fluid problems in the same manner. Therefore, it has also been applied to FSI problems such as membrane-gas interactions by York et al. [1999], and water-soil interactions by Lee and Guilkey [2008].

Recently, many efforts have been made to improve the material point method, such as the refinement of the contact algorithm by Bardenhagen et al. [2001] and the use of Generalized Interpolation Material Point (GIMP) method proposed by Bardenhagen and Kober [2004]. Despite all these improvements, there were still gaps to be filled in order to use MPM or GIMP for solving FSI problems induced by surface tension. First, a systematic study of GIMP for fluid dynamics was not found in the literature. Second, there was no surface tension model implemented in GIMP before this work. Third, the existing contact algorithm for GIMP does not consider the effect of surface tension and contact angle.

## 1.2 Objectives

The ultimate goal of this study is to extend the GIMP method for modeling fluid-solid interactions induced by surface tension, providing a single computational framework that is used for both the fluid and solid phases. To achieve this goal, the objectives are as follows:

1. to investigate the capability of GIMP in modeling fluid dynamics in a systematic way;
2. to develop a surface tension model in GIMP;
3. to develop a method to solve fluid-solid interactions induced by surface tension in GIMP.

Since accurate modeling of the behavior of the fluid phase is important to model FSI problems, systematic investigations of the GIMP method in fluid mechanics are first conducted. A wide range of fluid mechanics problems are studied in GIMP, which are presented in Chapter 2. The results and findings there enable us to better understand both

the advantages and limitations of GIMP in the modeling and simulations of fluids. Chapter 2 also reports special techniques that can be used to reduce errors in GIMP for fluid dynamics. The work presented in Chapter 2 is referred to as GIMP-CFD.

In Chapter 3, we expand the modeling capability of the GIMP method to include surface tension. An algorithm is developed to formulate the surface tension force in GIMP by using the CSF method and named as GIMP-CSF. The newly developed algorithm is verified by a few numerical examples which are compared to analytical solutions. The results show that our algorithm is effective in modeling surface tension with the GIMP method. This Chapter has been published in *Computer Modeling in Engineering and Sciences* as Chen et al. [2012].

In Chapter 4, a new method, which is named as GIMP-FSI, is presented for solving FSI problems dominated by surface tension based on the GIMP method. The method takes advantage of both GIMP-CFD (presented in Chapter 2) and GIMP-CSF (presented in Chapter 3). Compared to GIMP-CSF which is for fluids only, GIMP-FSI considers the surface tension effect as well as the contact angle effect for both fluid and solid. GIMP-CSF also adopts a new contact algorithm that allows partial slip of fluid at the triple points. Numerical examples are presented where three different examples are used to verify the proposed algorithm.

Finally, Chapter 5 summarizes the key achievements and findings of this thesis and discusses potential future work in using GIMP for FSI problems.



### 1.3 References

- S. Bardenhagen and E. Kober. The generalized interpolation material point method. *CMES: Computer Modeling in Engineering and Sciences*, 5:477–495, 2004.
- S. Bardenhagen, J. Guilkey, K. Roessig, J. Brackbill, W. Witzel, and J. Foster. An improved contact algorithm for the material point method and application to stress propagation in granular material. *CMES: Computer Modeling in Engineering and Sciences*, 2:509–522, 2001.
- Steffen Basting and Martin Weismann. A hybrid level set–front tracking finite element approach for fluid–structure interaction and two–phase flow applications. *Journal of Computational Physics*, 255(0):228 – 244, 2013.
- J. U. Brackbill, D.B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100:335–354, 1992.
- J.U. Brackbill and H.M. Ruppel. FLIP: a method for adaptively zoned, particle-in-cell calculation of fluid flows in two dimensions. *Computer Physics Communications*, 95:314–343, 1986.
- L. Chen, J. H. Lee, and C.-f. Chen. On the modeling of surface tension and its applications by the generalized interpolation material point method. *CMES: Computer Modeling in Engineering and Sciences*, 86:199–223, 2012.
- A.K. Das and P.K. Das. Equilibrium shape and contact angle of sessile drops of different volumes-computation by SPH and its further improvement by DI. *Chemical Engineering Science*, 65(13):4027 – 4037, 2010.
- A.J. Gil, A. Arranz Carreño, J. Bonet, and O. Hassan. An enhanced immersed structural potential method for fluid–structure interaction. *Journal of Computational Physics*, 250(0): 178 – 205, 2013.
- Nikil Kapur and Philip H. Gaskell. Morphology and dynamics of droplet coalescence on a surface. *Phys. Rev. E*, 75:056315, May 2007.
- M. Lahooti, A. Pishavar, and M.S. Saidi. A novel 2D algorithm for fluid solid interaction based on the smoothed particle hydrodynamics (SPH) method. *Scientia Iranica*, 18(3):358 – 367, 2011.

- J. H. Lee and J. E. Guilkey. Grain-scale modeling of saturated soils using the generalized interpolation material point method. In *Proceedings of the 16th International Conference of the International Society for Terrain-Vehicle Systems*, 2008.
- JaeHo Shin, JiSeok Lee, and SangHwan Lee. A strain-rate model for a lattice Boltzmann BGK model in fluid–structure interactions. *Computers and Fluids*, 88:126 – 135, 2013.
- D. Sulsky, Z. Chen, and H.L. Schreyer. A particle method for history dependent materials. *Computer Methods in Applied Mechanics and Engineering*, 118:179–196, 1994.
- D. Sulsky, S.J. Zhou, and H.L. Schreyer. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications*, 87:236–252, 1995.
- Sheng Xu and Z. Jane Wang. A 3D immersed interface method for fluid–solid interaction. *Computer Methods in Applied Mechanics and Engineering*, 197(25-28):2068 – 2086, 2008.
- A. R. York, D. L. Sulsky, and H. L. Schreyer. The material point method for simulation of thin membranes. *International Journal for Numerical Methods in Engineering*, 44:1429–1456, 1999.
- Thomas Young. An essay on the cohesion of fluids. *Philosophical Transactions of the Royal Society of London*, 95:pp. 65–87, 1805.



## Chapter 2

### Using the Generalized Interpolation Material Point Method for Fluid Dynamics at Low Reynolds Numbers<sup>1</sup>

#### 2.1 Abstract

The generalized interpolation material point (GIMP) method is a finite-element-like particle method using both the Lagrangian particles and Eulerian computational meshes for solving large-deformation problems in solid mechanics. In this paper, the capability of GIMP is extended to computational fluid dynamics (CFD) for flows at low Reynolds numbers. GIMP-CFD is systematically benchmarked by five validation problems: hydrostatic pressure, Rayleigh's problem, Poiseuille flows, lid-driven cavity flow, and dam-break employing an equation of state for water. A reduced bulk modulus was used to speed up computation. Pressure smoothing and more particles per cell were employed for problems with large pressure fluctuations. A comparison of the results using GIMP with those using the B-spline interpolation function showing that GIMP has better performance than B-spline. Evaluation of GIMP-CFD was conducted by comparing numerical solutions against known solutions showing good agreement. Our work indicates that GIMP-CFD is a promising method for fluid dynamics at low Reynolds numbers.

#### 2.2 Introduction

Mesh-free, meshless and particle methods [Li and Liu, 2002, Nguyen et al., 2008] have gained popularity in recent years to address issues, such as mesh entanglement in large deformations and fragmentation of a body, that are difficult to handle by traditional computational methods such as the finite element method.

Developed by Sulsky et al. [1994, 1995], MPM is a relatively new particle method which extends the hydrodynamics fluid implicit particle (FLIP) method [Brackbill and Ruppel, 1986] to solid mechanics. Like many other particle methods, MPM discretizes the solid body into a collection of particles (the material points). One unique feature of MPM is the adoption of a background mesh as a computational 'scratch pad' to deal with impact/contact and penetration types of problems [Chen and Brannon, 2002]. The background mesh, or grid, is discretized into a lattice of cells (elements) and grid points (nodes). The background mesh at the end of a time step is deformed, but is usually reset before the

---

<sup>1</sup>This chapter will be submitted for publication as L. Chen, J. H. Lee, and C. -f. Chen. Using the Generalized Interpolation Material Point Method for Fluid Dynamics at Low Reynolds Numbers.

next time step. Consequently, MPM can be considered as a hybrid finite-element-like particle method and has been contrasted with the finite element method [Chen and Brannon, 2002] and the popular smoothed-particle hydrodynamics (SPH) method [Ma et al., 2009]. To improve the accuracy and stability of MPM, Bardenhagen and Kober [2004] developed the generalized interpolation material point method (GIMP), a variant of MPM. Analysis and implementation choices of MPM have been reported in Steffen et al. [2008a,b].

Since its inception, MPM has been mainly applied to solve solid mechanics problems; its application to fluid dynamics is, however, limited. For example, Chen et al. [2012] first implemented a surface tension model using GIMP. Membrane-gas interaction problems involving gas dynamics were simulated by York et al. [1999]. Water-soil interactions were investigated by Lee and Guilkey [2008]. The last two works are examples of fluid-structure interaction (FSI) problems which can be challenging [Hou et al., 2012]. Extending MPM to fluid dynamics, in addition to its own merit, has the potential of solving FSI problems within a single computational framework as opposed to using a coupled approach [Guilkey et al., 2007]. To the best of our knowledge, no systematic studies of MPM for fluid dynamics with validation have been reported in literature. The purpose of this paper is to extend the capability of MPM to CFD using GIMP.

The rest of the paper is organized as follows. In Section 2.3, we first review MPM and then introduce the numerical procedure of MPM-CFD to handle fluid dynamics problems. In Section 2.4, we apply MPM-CFD to five benchmark examples and compare the results with known solutions. In Section 2.5, we discuss our numerical experience in using MPM-CFD for a balance between computational efficiency and accuracy, and draw conclusions.

## 2.3 MPM for fluid dynamics

In this section, essential elements of MPM are first reviewed. We then discuss the extension of MPM for fluid dynamics problems.

### 2.3.1 MPM formulations

MPM and its variants share a common high-level algorithm which is briefly reviewed here. Conservation of mass and momentum equations are shown, respectively, as:

$$\frac{d\rho}{dt} + \rho \nabla \cdot \mathbf{v} = 0 \quad (2.1)$$

and

$$\rho \mathbf{a} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} \quad (2.2)$$

where  $\rho$  is the density,  $\mathbf{v}$  is the velocity,  $\boldsymbol{\sigma}$  is the stress tensor,  $\mathbf{a}$  is the acceleration, and  $\mathbf{b}$  is the specific body force. For fluid dynamics problems where surface tension or wetting phenomenon is present, surface tension can also be modeled as a body force [Chen et al., 2012].

In terms of spatial discretization, MPM uses a finite number of particles to represent the object of interest. The positions of the particles,  $\mathbf{x}_p$ , are tracked throughout computation; the subscript  $p$  denotes particles. In MPM, the particles carry all the state variables and material properties including the mass  $M_p$ , velocity  $\mathbf{v}_p$ , stress  $\boldsymbol{\sigma}_p$ , and deformation gradient  $\mathbf{F}$ . Since the mass of each particle remains unchanged over time, conservation of mass (Eq. 2.1) is automatically satisfied.

Besides the Lagrangian particles, MPM uses a background mesh as a scratch pad to solve the momentum equation (Eq. 2.2). The acceleration on the  $i$ -th grid node is solved using:

$$m_i \mathbf{a}_i = \mathbf{f}_i^{ext} - \mathbf{f}_i^{int} \quad (2.3)$$

which is derived from the weak form of Eq. 2.2;  $m_i$  denotes the grid mass,  $\mathbf{a}_i$  is grid acceleration,  $\mathbf{f}_i^{ext}$  represents the grid external force, and  $\mathbf{f}_i^{int}$  is the grid internal force. Each of the aforementioned grid variables is in turn obtained by mapping particle information via an interpolation function  $S_{ip}$ :

$$m_i = \sum_p S_{ip}(\mathbf{x}_p) M_p \quad (2.4)$$

$$\mathbf{f}_i^{int} = M_p \sum_p \mathbf{G}_i(\mathbf{x}_p) \cdot \boldsymbol{\sigma}_p^s \quad (2.5)$$

$$\mathbf{f}_i^{ext} = \sum_p S_{ip}(\mathbf{x}_p) \mathbf{b}_p + \sum_p S_{ip}(\mathbf{x}_p) \boldsymbol{\tau}_p \quad (2.6)$$

where  $\sum_p$  denotes summation over particles,  $\mathbf{G}_i(\mathbf{x}_p) = \nabla S_{ip}$  is the gradient of the interpolation function,  $\boldsymbol{\sigma}_p^s$  is the specific stress tensor,  $\mathbf{b}_p$  is particle body force, and  $\boldsymbol{\tau}_p$  is particle traction force.

There exist many options for the interpolation function  $S_{ip}$  including the linear function (used in the original MPM), the GIMP interpolation function [Bardenhagen and Kober, 2004], and the B-spline interpolation function [Steffen et al., 2008a,b]. In this paper, we pri-

marily used the GIMP interpolation function because it is computationally robust for general applications. A B-spline interpolation function, reported to show good performance [Steffen et al., 2008a], was also used for comparison.

After Eq. 2.3 is solved for the acceleration  $\mathbf{a}_i$  on the grid nodes, the new grid velocity,  $\mathbf{v}_i^L$ , can be calculated for a time increment  $\Delta t$ :

$$\mathbf{v}_i^L = \mathbf{v}_i + \mathbf{a}_i \Delta t \quad (2.7)$$

in which  $\mathbf{v}_i$  is the old grid velocity given by

$$\mathbf{v}_i = \frac{\sum_p S_{ip}(\mathbf{x}_p) \mathbf{v}_p M_p}{m_i} \quad (2.8)$$

Then the grid acceleration and the new grid velocity are mapped (interpolated) to particles for the next computational cycle. The incremental particle velocity is given by

$$\Delta \mathbf{v}_p = \sum_i S_{ip} \mathbf{a}_i \Delta t \quad (2.9)$$

and the incremental particle position is given by

$$\Delta \mathbf{x}_p = \sum_i S_{ip} \mathbf{v}_i^L \Delta t \quad (2.10)$$

The particle stress is updated next using the update-stress-last (USL) algorithm which was reported to outperform the update-stress-first (USF) algorithm [Wallstedt and Guilkey, 2008]. The increment of particle stress,  $\Delta \sigma_p$ , is obtained as:

$$\Delta \mathbf{F}_p = \mathbf{I} + \Delta t \sum_i \mathbf{G}_{ip} \mathbf{v}_i^L \quad (2.11)$$

and

$$\Delta \sigma_p = f(\Delta \mathbf{F}_p) \quad (2.12)$$

where  $\Delta \mathbf{F}_p$  is the increment of the deformation gradient tensor,  $\mathbf{I}$  the identity tensor, and  $f(\Delta \mathbf{F}_p)$  describes the constitutive law of the material.

For explicit time integration of a dynamic problem, artificial damping is often used to increase the energy dissipation to speed up computation. In MPM, artificial damping can

be implemented by adding a damping term to the grid acceleration:

$$\mathbf{a}_i^{damp} = \mathbf{a}_i - \alpha \mathbf{v}_i^L \quad (2.13)$$

where  $\alpha$  is the damping coefficient chosen by the user suitable for the problem at hand.

### 2.3.2 Constitutive model for fluid

In MPM, a fluid material is treated just like a solid material but adopting a different constitutive model. For a nearly-incompressible fluid, we adopt the constitutive law commonly used in SPH and given in Cueto-Felgueroso et al. [2004] to calculate the stress:

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\mathbf{d}' \quad (2.14)$$

where  $\mu$  is the dynamic viscosity,  $\mathbf{d}'$  is the rate of deformation tensor based on  $\Delta\mathbf{F}_p$ , and  $p$  is the hydrostatic pressure determined by an equation of state given by Monaghan [1994] and Morris et al. [1997]:

$$p = K \left[ \left( \frac{\rho}{\rho_0} \right)^\gamma - 1 \right] \quad (2.15)$$

where  $K$  is the bulk modulus,  $\rho$  the fluid density,  $\rho_0$  the initial density, and  $\gamma$  a constant (e.g., 7.0 for water and 1.4 for air under room temperature and atmospheric pressure). We use  $\gamma=7.0$  for all the examples in Section 2.4.

For time integration of explicit dynamic problems, the Courant-Friedrichs-Lewy (CFL) condition needs to be satisfied for stability:

$$\Delta t_{cr} = \frac{dx}{c} \quad (2.16)$$

where  $\Delta t_{cr}$  is the critical time step,  $dx$  the minimum cell spacing, and  $c$  the speed of sound:

$$c = \sqrt{\frac{K}{\rho}} \quad (2.17)$$

The bulk modulus can be very high for a nearly incompressible fluid, such as water, resulting in a very small time step. It is suggested that a reduced bulk modulus be used for such a fluid as long as the change in density is less than 3% [Monaghan, 1994, Staroszczyk, 2010].



In practice, the size of time step is a fraction of the value given in Eq. 2.17, typically 0.2-0.4, to obtain desired results for the problem at hand. The stable time step size is updated every computational cycle based on Eq. 2.16 and the chosen fraction.

### 2.3.3 Pressure stabilization

Pressure oscillation can occur for an incompressible fluid modeled using Eq. 4.37 in conjunction with a high bulk modulus. A pressure smoothing technique, similar to conventional finite element method for incompressible materials [Chen et al., 1995, Lee et al., 1979], can be used to mitigate this oscillation. In this paper, we use a local-cell smoothing scheme in which the particle pressure is recalculated by obtaining an averaged Jacobian of the deformation gradient  $\mathbf{F}$  using the following equation:

$$J_p^* = \frac{\sum_{p=1}^N J_p V_p}{\sum_{p=1}^N V_p} \quad (2.18)$$

where  $J_p$  is the Jacobian of the deformation gradient tensor  $\mathbf{F}$ ,  $N$  the number of particles in one cell,  $V_p$  the particle volume, and  $J_p^*$  the averaged Jacobian. Note that  $J_p = \rho/\rho_0$ .

## 2.4 Numerical examples and validations

In this section, five common examples for validating CFD codes are presented to demonstrate the capability and limitation of our approach described in Section 2.3. The examples are: the hydrostatic pressure of a water column subjected to gravity, the transient behavior of viscous flow for Rayleigh's problem, two-dimensional and three-dimensional Poiseuille flows, the lid-driven cavity problem, and a simplified dam-break problem. In this paper, we focus on flows at low Reynolds numbers.

To quantify numerical errors, we use the  $L_\infty$  norm, defined as:

$$L_\infty = \frac{\max |x_{true} - x_{cal}|}{\max |x_{true}|}, \quad (2.19)$$

and the  $L_2$  norm, defined as:

$$L_2 = \sqrt{\frac{\sum (x_{true} - x_{cal})^2}{\sum x_{true}^2}} \quad (2.20)$$

where  $x_{true}$  is the theoretical solution, and  $x_{cal}$  is the numerical solution.

We use two interpolation functions, GIMP and B-spline, in all of the benchmark ex-

amples and compare their performance in terms of computational efficiency and accuracy. To differentiate current work with the traditional MPM developed for solid mechanics, the methodology presented here is called MPM-CFD in general, and GIMP-CFD when the GIMP interpolation function is used.

All the results are obtained using the open-source software package Uintah [Parker et al., 2006] running on a Sun Ultra 40 computer with a 2.4 GHz CPU.

### 2.4.1 GIMP-CFD

#### Hydrostatic pressure

Fluid at rest is a simple problem where the hydrostatic pressure is proportional to depth. Since its solution is independent of viscosity, the problem can be used to evaluate the equation of state specified in Eq. 4.37. The numerical error of GIMP-CFD can be compared to the theoretical solution per Pascal's law.

As illustrated in Fig. 2.1a, a two-dimensional water column in a rectangular tank is subject to gravitational acceleration ( $g=10 \text{ m/s}^2$ ). The distribution of pressure in the problem domain, represented by a straight line, is schematically plotted per Pascal's law. A  $0.04 \text{ m} \times 0.1 \text{ m}$  water column in a  $0.04 \text{ m} \times 0.15 \text{ m}$  tank is modeled. The problem domain is meshed into square cells where each cell has a length ( $dx$ ) of  $0.01 \text{ m}$ , one-tenth of the height of water ( $h$ ), i.e.,  $dx/h=0.1$ . The number of particles per cell (PPC) is  $2 \times 2$ . For this two-dimensional problem, only one particle and one cell are used in the third dimension where the symmetric boundary condition is applied; this approach is used for all two-dimensional examples in this paper. Symmetric boundary conditions are also used for the vertical sides of the domain. At the bottom of the domain, a fixed Dirichlet boundary condition is used.

The only external force is due to gravity activated at  $t=0 \text{ s}$ . The viscosity of water,  $0.001 \text{ Pa}\cdot\text{s}$ , is used. Two values of the bulk modulus are used:  $2.2 \times 10^9 \text{ Pa}$  representing water ( $K_w$ ), and a reduced modulus ( $K_r$ ) of  $1.54 \times 10^4 \text{ Pa}$  to speed up the computation per discussions in Section 2.3.2.

Figure 2.1b shows numerical results of particle pressure versus height (measured from the bottom of the tank) using the reduced modulus  $K_r$  with good comparison to theory. The results are based on  $t=0.2 \text{ s}$  when a steady state is reached using an artificial damping coefficient of 100. Note that the use of artificial damping does not affect the results since a

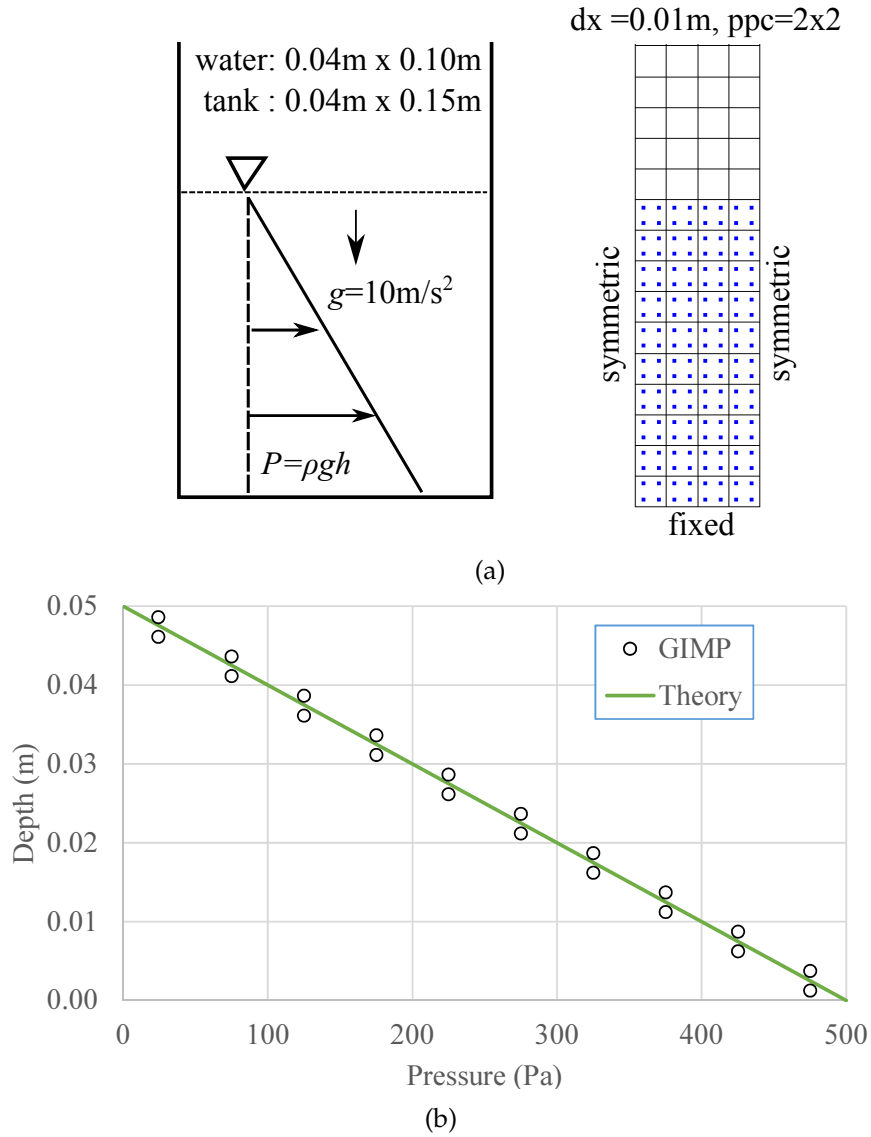


Figure 2.1: (a) Problem setup for hydrostatic pressure simulation (left), discretization of the domain (fluid particles and background mesh) with boundary conditions (right); (b) comparison of pressure vs. depth (measured from the bottom of the tank) at equilibrium state between theory and numerical results at the particles using the reduced bulk modulus  $K_r = 1.54 \times 10^4 \text{ Pa}$ .

steady-state solution is sought.

The  $L_\infty$  error is 2.5%, as opposed to the  $L_2$  error which is 6%. It indicates that GIMP-CFD performs well. For comparison, when  $K_w$  is used,  $L_\infty$  error remains at 2.5%, whereas the  $L_2$  error is slightly smaller than 4.1%. However, computation time using  $K_r$  is 18 seconds, and that using  $K_w$  is significantly higher at 1644 seconds.

We also study the effects of mesh size and number of particles per cell (PPC) which are summarized in Table 3.2 using the reduced modulus  $K_r$ . Generally, the numerical error decreases with a smaller mesh size, as expected. Increase in PPC slightly improves the accuracy.

In summary, whether using  $K_w$  or  $K_r$ , GIMP-CFD is effective in calculating the hydrostatic pressure of water.

### Rayleigh's problem

The second numerical example is Rayleigh's problem where the fluid motion is dominated by the viscous force. The problem is illustrated in the left figure of Fig. 2.2a where an initially stationary, viscous fluid is moving in the  $x$  direction due to an infinite rigid plate coinciding with the  $x$  axis and traveling at a constant velocity  $U$ . This problem is typically described as a one-dimensional flow in a 2D domain. The momentum equation is reduced to:

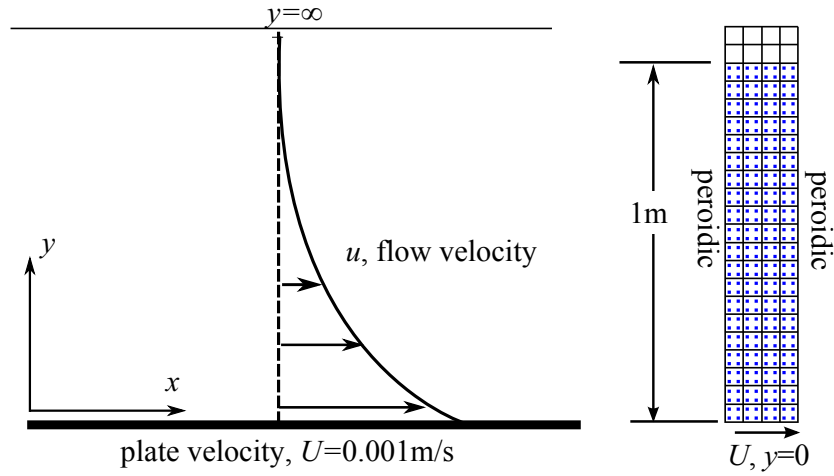
$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} \quad (2.21)$$

where  $u$  is the velocity of the flow in the  $x$  direction, and  $\nu$  the kinematic viscosity. Since the rigid plate is infinite in the  $x$  direction, the continuity equation is reduced to:

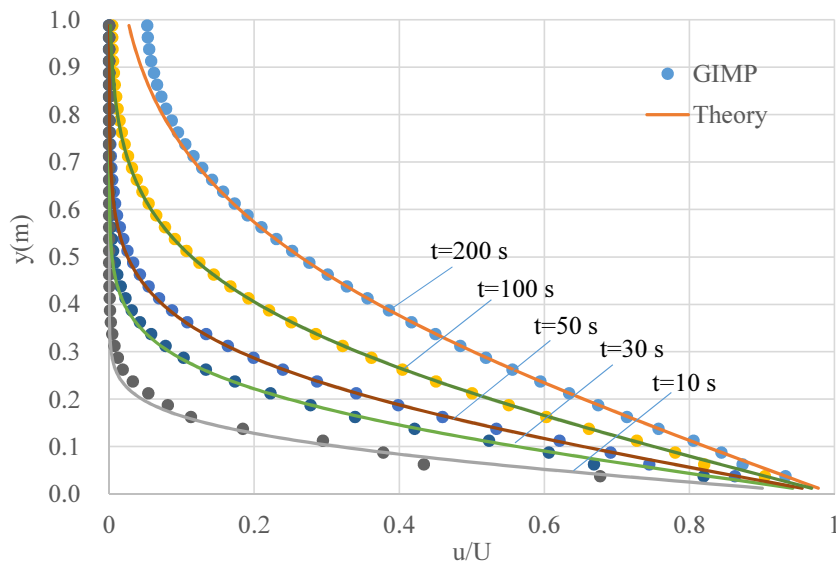
$$\frac{\partial v}{\partial y} = 0 \quad (2.22)$$

where  $v$  is the velocity in the  $y$  direction. The boundary conditions are  $u = 0$  at  $y = \infty$ , and  $u = U$  at  $y = 0$ . The initial conditions are  $u = v = 0$  at  $t = 0$ . Under these conditions,  $v$  must always be zero. The solution of  $u$  is mathematically identical to the well-known solution of a corresponding heat conduction problem for a semi-infinite rod [Carslaw and Jaeger, 1986]:

$$u = U(1 - \operatorname{erf} \frac{y}{2\sqrt{\nu t}}) \quad (2.23)$$



(a)



(b)

Figure 2.2: (a) Problem setup for Rayleigh's problem (left), discretization of the domain (fluid particles and background mesh) with boundary conditions (right); (b) calculated normalized velocity profile versus theoretical profile.

The right figure of Fig. 2.2a shows the setup of the problem where a two-dimensional fluid ( $0.2 \text{ m} \times 1.0 \text{ m}$ ) is modeled by 320 particles ( $8 \times 40$ ) in a  $0.2 \text{ m} \times 1.1 \text{ m}$  domain which has  $4 \times 22$  cells,  $\text{PPC}=2 \times 2$ . The bulk modulus of the fluid, assumed to have no influence on the results, is taken as  $1.5 \times 10^4 \text{ Pa}$ . Viscosity  $\nu$  is taken as  $0.0005 \text{ m}^2/\text{s}$ . The Dirichlet boundary condition is applied at the lower edge of the domain to simulate the moving plate with  $U=0.001 \text{ m/s}$ . Since the fluid flow is uniform along the  $x$  direction, a periodic boundary condition is applied at the left and right edges.

The comparison of the theoretical and simulation results of the normalized velocity  $u/U$  versus  $y$  at different times in Figure 2.2b shows a good agreement. Numerical error is the largest at  $t=10 \text{ s}$  with  $L_\infty=23.0\%$  and  $L_2=17.4\%$ , shown as the maximum values in the first row of Table 2.2. However, the error decreases with time: the smallest error is at  $t=200 \text{ s}$  with  $L_\infty=4.57\%$ , and  $L_2=2.41\%$ , shown as the minimum values in the first row of Table 2.2.

As shown in Table 2.2, the accuracy can be increased by using a refined mesh ( $4 \times 40$ ), as expected. Table 2.2 shows that an increase of PPC improves slightly the accuracy in  $L_2$ .

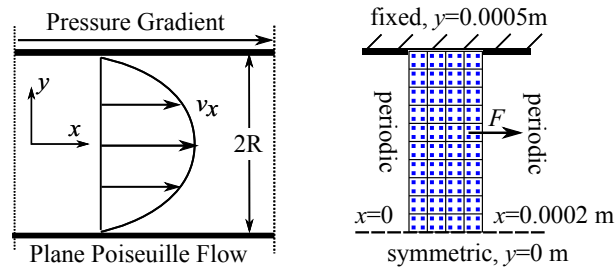
### Time-dependent Poiseuille flow

The Poiseuille flow is another typical problem where the viscous effect dominates the fluid behavior. Time-dependent Poiseuille flows, which were previously simulated by the SPH method [Sigalotti et al., 2003], are modeled in this paper. The theoretical solutions for both the two-dimensional and three-dimensional Poiseuille flows can be found in literature [Sigalotti et al., 2003] to benchmark the accuracy of GIMP-CFD.

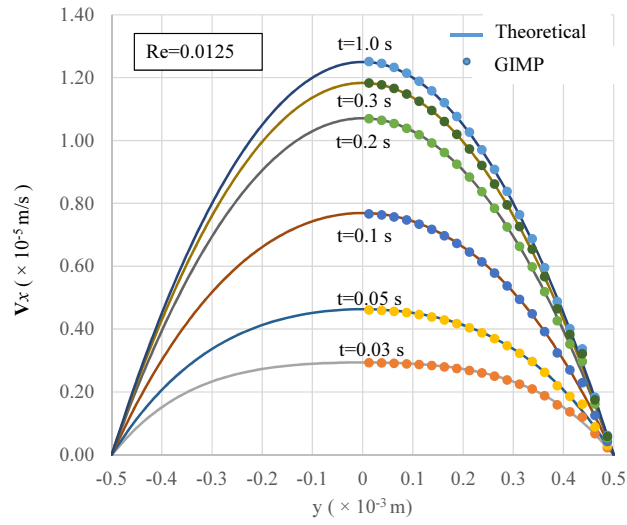
#### *Transient 2D Poiseuille flow*

As shown in Fig. 2.3a, the two-dimensional Poiseuille flow is bounded between two parallel plates fixed at  $y = -R$  and  $y = R$ , and driven by a constant pressure gradient in the  $x$  direction. The velocity in the  $x$  direction can be described by a series solution [Sigalotti et al., 2003]:

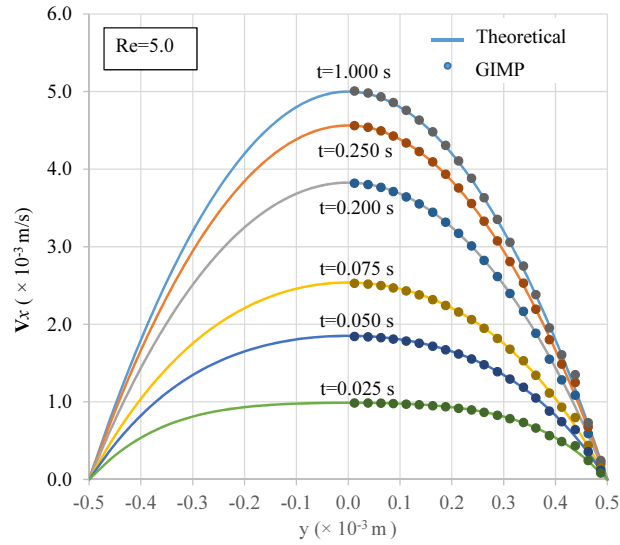
$$v_x(y, t) = \frac{F}{2\nu}(y^2 - R^2) + \sum_{n=0}^{\infty} \frac{16(-1)^n R^2 F}{\mu \pi^3 (2n+1)^3} \cos\left(\frac{(2n+1)\pi y}{2R}\right) \exp\left(\frac{-(2n+1)^2 \pi^2 \mu}{4R^2} t\right) \quad (2.24)$$



(a) Plane Poiseuille flow and GIMP model;  $F$  is the body force.



(b) Computed velocity profile versus theoretical profile for  $Re=0.0125$ .



(c) Computed velocity profile versus theoretical profile for  $Re=5$ .

Figure 2.3: Simulation of plane Poiseuille flow by GIMP.

where  $F$  is a constant body force that produces the needed pressure gradient to drive the flow,  $\nu$  the kinematic viscosity, and  $2R$  the distance between the plates. The maximum velocity at the steady state along  $y = 0$  is:

$$v_{max}(y = 0, t = \infty) = \frac{r^2 F}{2\nu}. \quad (2.25)$$

Similar to the SPH simulations in Sigalotti et al. [2003], we used two Reynolds numbers ( $Re$ ): 0.0125 and 5.0. The Reynolds number is calculated by  $Re = \frac{2Rv_{max}}{\nu}$ , or equivalently by  $Re = \frac{R^3 F}{\nu^2}$ . Therefore, for  $Re = 0.0125$ , we chose  $\nu = 10^{-6} m^2/s$ , and  $F = 10^{-4} m/s^2$ ; for  $Re = 5$ , we chose  $\nu = 10^{-6} m^2/s$ , and  $F = 0.04 m/s^2$ . The bulk modulus is  $1.5 \times 10^4$  Pa, same as used before.

Due to symmetry, only one half of the domain is modeled. As shown in Fig. 2.3a, a symmetric boundary condition is applied at  $y = 0$  with a fixed boundary condition at  $y = 0.0005$  m. Since the flow is uniform in the  $x$  direction, the periodic boundary condition is applied at the left and right edges of the domain. A  $4 \times 10$  mesh is used in the discretized domain with  $PPC=2 \times 2$ .

The flow velocity profiles in the  $y$  direction at different times are plotted against the theoretical solutions in Fig. 2.3b ( $Re = 0.0125$ ) and Fig. 2.3c ( $Re = 5$ ). Both flows eventually reach the steady state at  $t=1.0$  s.

Table 2.3 summarizes the effects of mesh size and PPC on the simulation results. For both  $Re = 0.0125$  and  $Re = 5$ ,  $L_\infty$  ranges from 3.5% to 5.6%, and  $L_2$  from 1.1% to 1.7% at a grid of  $4 \times 10$  and  $PPC=2 \times 2$ . These numerical errors provide evidence of the effectiveness of GIMP-CFD. Similar to the results of the hydrostatic pressure example and Rayleigh's problem, the mesh size has a larger influence on accuracy than PPC.



Table 2.1: Effects of mesh size and PPC for the hydrostatic problem expressed in  $L_\infty$  and  $L_2$  using the reduced bulk modulus  $K_r=1.54 \times 10^4$  Pa.

$dx/h$	PPC	$L_\infty$ (%)	$L_2$ (%)
0.1	$2 \times 2$	2.50	6.00
0.2	$2 \times 2$	5.50	8.50
0.05	$2 \times 2$	1.40	2.20
0.1	$3 \times 3$	3.50	4.60
0.1	$4 \times 4$	3.80	4.70

Table 2.2: Effects of mesh size and PPC on Rayleigh's problems. For each case, minimum and maximum of  $L_\infty$  and  $L_2$  are given.

Grid	PPC	$L_\infty$ (min, max)%	$L_2$ (min, max)%
$4 \times 20$	$2 \times 2$	(4.57, 23.0)	(2.41, 17.4)
$4 \times 40$	$2 \times 2$	(2.51, 10.2)	(0.98, 5.74)
$4 \times 10$	$2 \times 2$	(9.42, 45.0)	(5.30, 40.4)
$4 \times 20$	$3 \times 3$	(4.60, 22.6)	(2.50, 15.5)
$4 \times 20$	$4 \times 4$	(4.67, 22.7)	(2.25, 14.9)

Table 2.3: Effects of mesh size and PPC on plane Poiseuille flow using GIMP.

Re=0.0125			
Grid	PPC	$L_\infty$ (min, max)%	$L_2$ (min, max)%
$4 \times 5$	$2 \times 2$	(7.37, 7.78)	(3.25, 4.20)
$4 \times 10$	$2 \times 2$	(3.52, 5.57)	(1.14, 1.69)
$4 \times 20$	$2 \times 2$	(1.81, 3.27)	(0.42, 0.67)
$4 \times 10$	$3 \times 3$	(3.20, 5.72)	(1.35, 2.12)
$4 \times 10$	$4 \times 4$	(3.15, 5.59)	(1.11, 1.79)
Re=5			
$4 \times 5$	$2 \times 2$	(7.37, 7.77)	(3.25, 4.25)
$4 \times 10$	$2 \times 2$	(3.52, 5.57)	(1.14, 1.69)
$4 \times 20$	$2 \times 2$	(1.71, 3.12)	(0.40, 0.64)
$4 \times 10$	$3 \times 3$	(3.20, 5.82)	(1.35, 2.12)
$4 \times 10$	$4 \times 4$	(3.15, 5.59)	(1.11, 1.79)

### Transient 3D Poiseuille flow

Figure 2.4 shows a schematic of the Poiseuille flow in three dimensions. Due to axisymmetry, the flow can be described in the radial and axial coordinates  $(r, z)$  only. Similar to the plane Poiseuille flow, the flow is driven by a pressure gradient, modeled by a body force  $F$ , in the  $z$  direction. According to Sigalotti et al. [2003], the flow velocity  $v_z(r, t)$  across the radius at time  $t$  can be expressed in an analytical form:

$$v_z(r, t) = \frac{F}{4\mu}(r^2 - R^2) + \sum_{m=1}^{\infty} \frac{FR^2}{\nu\alpha_m^2} \frac{J_2(\alpha_m)}{J_1^2(\alpha_m)} J_0\left(\frac{r\alpha_m}{R}\right) \exp\left(-\frac{\nu\alpha_m^2 t}{R^2}\right) \quad (2.26)$$

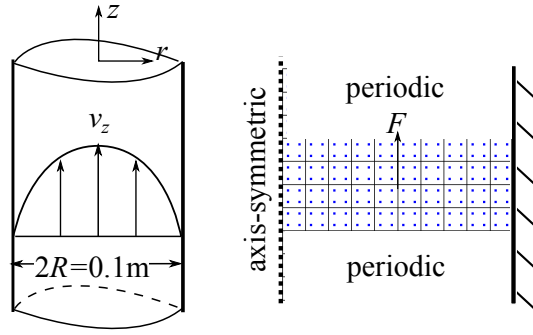
where  $J_0$ ,  $J_1$ , and  $J_2$  are Bessel functions of the first kind at different orders, and  $\alpha_m$  is the  $m$ -th roots of  $J_0$ . The maximum velocity at the steady state is given by:

$$v_{max} = \frac{R^2 F}{4\nu} \quad (2.27)$$

The Reynold's number can be calculated as  $\frac{2Rv_{max}}{\nu}$ , or equivalently as  $\frac{R^3 F}{2\nu^2}$ .

This 3D Poiseuille flow problem was simulated using the axisymmetric form of GIMP formulated by Narin and Guilkey [2013]. Fig. 2.4a shows the GIMP model of the problem for a pipe with a radius of 0.05 m discretized into a  $10 \times 4$  mesh, PPC=2×2. A symmetric boundary condition is applied at the left edge of the domain ( $r = 0$ ) due to axisymmetry. The right edge is fixed with a no-slip Dirichlet boundary condition. Similar to the plane Poiseuille flow, periodic boundary conditions are used in the direction of flow. Two Reynold's numbers are modeled:  $Re = 0.0125$ , and  $Re = 5.0$ . We chose  $\nu = 0.001 \text{ m}^2/\text{s}$  and  $F = 0.0002 \text{ m/s}^2$  for  $Re = 0.0125$ , and  $\nu = 0.0005 \text{ m}^2/\text{s}$  and  $F = 0.02 \text{ m/s}^2$  for  $Re = 5.0$ .

The velocity profiles as a function of time are plotted against the theoretical solutions in Fig. 2.4b ( $Re = 0.0125$ ), and Fig. 2.4c ( $Re = 5.0$ ). The results show that the axisymmetric form of GIMP is effective in simulating the 3D Poiseuille flow problem. The numerical errors from various mesh sizes and PPC are also investigated and summarized in Table 2.4. The  $L_{\infty}$  and  $L_2$  are both smaller than 5% when a finer mesh is used. Based on Table 2.4, the effects of PPC on the 3D Poiseuille flow are not as evident as mesh refinement.



(a) 3D Poiseuille flow and GIMP setup.

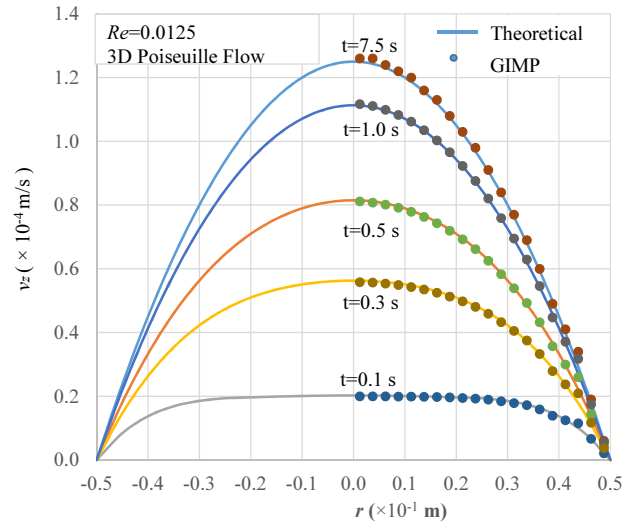
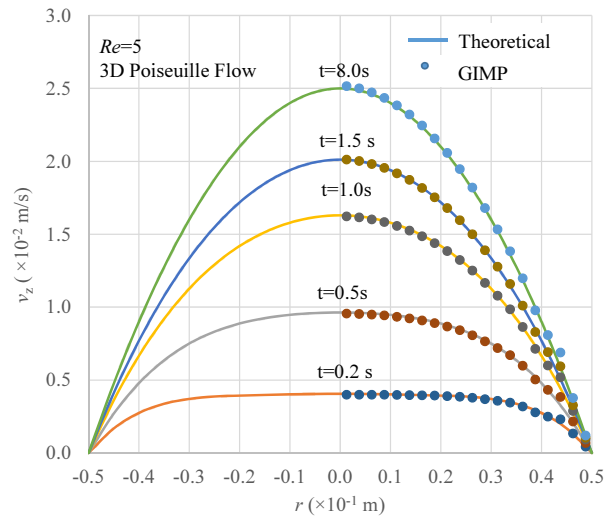
(b) Computed velocity profiles (dotted) and theoretical values (line) for  $Re = 0.0125$ .(c) Computed velocity profiles (dotted) and theoretical values (line) for  $Re = 5.0$ .

Figure 2.4: GIMP simulation of transient 3D Poiseuille flow in a pipe.

Table 2.4: Effects of mesh size and PPC on 3D Poiseuille flow using GIMP.

Re=0.0125			
Grid	PPC	$L_\infty(\text{min, max})\%$	$L_2(\text{min, max})\%$
4×5	2×2	(8.70, 10.0)	(4.60, 6.40)
4×10	2×2	(4.10, 7.80)	(1.40, 2.50)
4×20	2×2	(1.90, 4.70)	(0.50, 1.10)
4×10	3×3	(3.80, 8.70)	(1.60, 3.00)
4×10	4×4	(3.80, 7.90)	(1.30, 2.50)
Re=5			
4×5	2×2	(7.60, 9.80)	(4.40, 6.40)
4×10	2×2	(4.00, 7.50)	(1.40, 2.50)
4×20	2×2	(1.90, 4.70)	(0.46, 1.10)
4×10	3×3	(3.80, 8.70)	(1.60, 3.00)
4×10	4×4	(3.70, 7.80)	(1.30, 2.50)

### Lid-driven cavity flow

The validation problems we have presented so far show that GIMP-CFD can accurately model typical fluid dynamics problems where pressure follows a simple pattern, either constant or linear. A more challenging problem is the lid-driven cavity problem [Shankar and Deshpande, 2000] which has long been used to validate new CFD schemes since many complicated flow patterns can be obtained within a well-defined domain for this problem.

This two-dimensional problem is illustrated in Fig. 2.5a where a square cavity of side length  $L$  is filled with an incompressible fluid driven by a moving lid with a velocity  $U$  on the top of the cavity. Numerical and experimental studies show that the flow will become steady in the low Reynolds number regime ( $Re = \frac{UL}{\nu}$ ). Ghia et al. [1982] provided the solutions for Reynolds numbers between 100-1000. In this paper,  $Re = 100$  is used:  $\nu=0.01 \text{ m}^2/\text{s}$ ,  $U=10 \text{ m/s}$ , and  $L=0.1 \text{ m}$ . In Fig. 2.5a, a  $20 \times 20$  mesh is used for the domain with  $\text{PPC}=4 \times 4$  resulting in a total of 6,400 particles.

In order to yield reasonable numerical results using GIMP, four modeling choices have been considered: bulk modulus, PPC, boundary conditions, and pressure stabilization.

First, similar to the numerical examples presented before, a reduced bulk modulus ( $1.5 \times 10^6 \text{ Pa}$ ) is used here to prevent the calculation from diverging due to the very large pressures around the corners of the cavity. Using a reduced bulk modulus makes the

simulation stable while reducing the computation time.

The second aspect is that voids and clumped particles (i.e., particles that are very close to one another due to fluid flow) can be formed after a certain time of computation. This phenomenon has also been reported using the SPH method [Ma et al., 2009]. In GIMP-CFD, we found that the issue can be resolved by using more particles per cell. As shown in Fig. 2.5b, using  $PPC=4 \times 4$  yields far fewer voids and clumped particles than  $PPC=2 \times 2$ , and also gives a better approximation to the velocity field (shown as vectors).

The third aspect is that a minor modification of the boundary conditions for the lid-driven cavity problem can effectively mitigate the so-called singularity issue caused by the intersection of two different boundary conditions at the two corners of the top edge. For this purpose, the boundary condition at the top edge is modified to simulate the moving lid; as shown in Fig. 2.5a, only 80% of the top boundary, away from the corners, is allowed to move at the given speed.

The fourth aspect is the use of a pressure stabilization scheme (Eq. 2.18) to improve the results. As demonstrated in Fig. 2.5c, with pressure stabilization, the flow can quickly reach the steady state (after  $t=0.1$  s), which cannot be achieved without pressure stabilization. The  $x$ -component of the flow velocity (at  $x=0$ ) is plotted against the data from Ghia et al. [1982] at  $t=0.2$  s. It can be seen that GIMP with pressure stabilization gives much closer results to the experimental data than those without. The effectiveness of pressure stabilization may be due to the alleviation of the high degree of pressure fluctuation and distortion in the lid-driven cavity problem, both are not handled well in the original GIMP method without pressure stabilization.

With the considerations described above, GIMP-CFD is able to yield results that match well with experimental data (Fig. 2.5c).

### Breaking of a dam

To further demonstrate the capability of GIMP-CFD, a dam-break problem is considered here. Similar simulations using SPH can be found in Lee et al. [2008] and Staroszczyk [2010]. Since theoretical solutions are not available for the problem, numerical solutions from OpenFOAM [2013] are used as reference.

Figure 2.6 shows the geometry of a two-dimensional dam-break problem similar to the simulation by Rivola [2007] by OpenFOAM. A  $0.146 \text{ m} \times 0.292 \text{ m}$  fluid column rests at the left corner of a  $0.584 \text{ m} \times 0.584 \text{ m}$  container. The fluid column starts to collapse due

to a downward gravitational force ( $g=9.81 \text{ m/s}^2$ ) from  $t=0 \text{ s}$ . The flow is simulated using GIMP-CFD and OpenFOAM with the same fluid properties: viscosity  $\mu = 0.001 \text{ Pa}\cdot\text{s}$ , and density  $\rho = 1000 \text{ kg/m}^3$ . In OpenFOAM, the incompressible fluid is modeled while in GIMP-CFD a nearly incompressible fluid is modeled using a bulk modulus of  $1.5 \times 10^6 \text{ Pa}$ . For OpenFOAM, a  $100 \times 100$  mesh is used to discretize the problem domain. For GIMP-CFD simulation, a  $50 \times 50$  mesh is used as the computational domain; a total of 5000 particles are used to represent the fluid column with  $\text{PPC}=4 \times 4$ . In both methods, slip boundary conditions are used for all faces. For GIMP-CFD, slip boundary conditions are achieved using symmetric boundary conditions.

Similar to the lid-driven cavity problem, pressure stabilization is applied to obtain a smooth flow in the dam-break problem. Figs. 2.7a-2.7c show similar results obtained by OpenFOAM and GIMP-CFD at different times. For the dam-break problem, a PPC of  $4 \times 4$  is slightly helpful in reducing voids when compared with a PPC of  $2 \times 2$ . It is also found that pressure stabilization becomes crucial when a bulk modulus higher than  $1.5 \times 10^6 \text{ Pa}$  is used.

### Computation time

Table 2.5 shows the computation time for some of the numerical examples using GIMP-CFD. For Poiseuille flow simulation, only the cases with  $Re=0.125$  are listed since the cases with  $Re=5.0$  have similar computation time. For hydrostatic pressure problems, using artificial damping and reduced bulk modulus speed up the computation; the CPU time needed to reach stable pressures is generally less than a minute. For Rayleigh's problems, increasing grid resolution and PPC results in more computation time, as expected; with the same number of total particles (e.g., 1280), a finer mesh (e.g.,  $dx = 0.025 \text{ m}$ ) requires more CPU time than a coarser mesh (e.g.,  $dx = 0.05 \text{ m}$ ). Interestingly, simulation of the 3D Poiseuille flow using the axisymmetric form of GIMP requires much less time than the 2D Poiseuille flow, even though the grid resolution, PPC, and the time step size for both 2D and 3D cases are similar. Thus the axisymmetric form of MPM seems to be quite efficient in simulating the 3D Poiseuille flow. The computation time for the lid-driven cavity and dam-break problems is also listed. For the dam-break problem, GIMP-CFD requires much more time (5088 s) than OpenFoam (185 s), which is mainly due to the need in GIMP-CFD to map variables between particles and the background mesh, as well as due to the smaller time steps used in GIMP-CFD.

Table 2.5: Summary of computation time using GIMP-CFD.

Problem	dx (m)	PPC	Particles	CPU time (s)	Physical time (s)
Hydrostatic pressure	0.005	2×2	160	18	0.2
	0.005	3×3	360	16	0.2
	0.005	4×4	640	21	0.2
	0.0025	2×2	640	36	0.2
Rayleigh's problem	0.05	2×2	320	9500	200
	0.05	3×3	720	17979	200
	0.05	4×4	1280	19655	200
	0.025	2×2	1280	24200	200
2D Poiseuille Re=0.0125	0.005	2×2	160	9148	1
	0.005	3×3	360	12774	1
	0.005	4×4	640	17757	1
	0.0025	2×2	640	23046	1
3D Poiseuille Re=0.0125	0.005	2×2	160	668	8
	0.005	3×3	360	752	8
	0.005	4×4	640	1555	8
	0.0025	2×2	640	1501	8
Lid-driven Cavity	0.005	4×4	6400	1851	0.2
Dam-break	0.0117	4×4	5000	5088	1.0

#### 2.4.2 Alternative interpolation function – B-spline

To study the effect of interpolation function, results using the third-order B-spline interpolation function are shown in Figs. 2.8-2.10 for the same validation examples given earlier using GIMP.

For the hydrostatic problem (Fig. 2.8a), results using B-spline ( $dx/h=0.1$ , PPC=2×2) give similar results as GIMP away from the boundary, but show poor performance near the boundary. The  $L_2$  error from B-spline is 9.6% which is higher than that of GIMP (6.0%). The poor performance of B-spline near the boundaries is also found for Rayleigh's problems shown in Fig. 2.8b based on a grid of 4×40 and a PPC of 2×2. The maximum  $L_2$  is 15.5% which is higher than that of GIMP (5.74%). Furthermore, Fig. 2.9 shows that B-spline is ill-suited in simulating both the 2D and 3D Poiseuille flows with  $L_2$  ranging from 17% to 19% based on a grid of 4×10 and a PPC of 2×2, while GIMP with similar configurations gives  $L_2$  ranging from 1% to 4%.

For the problems of lid-driven cavity and dam break, using B-spline *without* using pressure stabilization, however, yields similar results as using GIMP with pressure stabilization as shown in Fig. 2.10. This could be due to the fact that B-spline is a smoother interpolation function than GIMP thus yields a similar smooth pressure field obtained by GIMP with pressure stabilization. Also, for the lid-driven cavity problem using the B-spline interpolation function, the effect of PPC is not significant as shown in Fig. 2.10.

## 2.5 Discussions and conclusions

In this paper, a numerical procedure extending MPM to fluid statics and dynamics problems was presented and validated by benchmarking five problems using analytical or well-accepted numerical or experimental solutions. These examples cover applications in hydrostatics, steady-state and transient 2D and 3D fluid flows in pipes, fluid dynamics problems with Reynolds numbers up to 100, and the dam-break problem which features very large deformations.

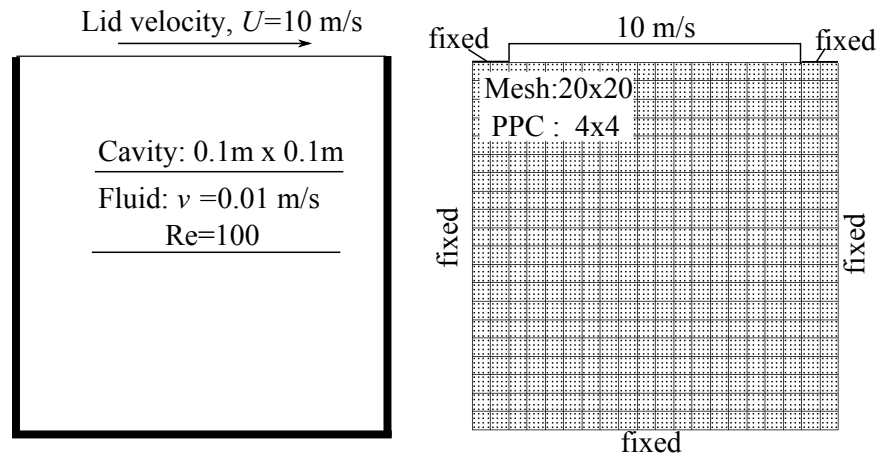
We have explored several options of MPM to obtain best results, balancing computational efficiency with accuracy. They include the choice of interpolation function (GIMP or B-spline), mesh size, particles per cell, pressure stabilization, artificial damping, and reduced bulk modulus. As in most numerical methods, mesh refinement leads to more accurate results. A higher PPC is useful for situations where voids or clumped particles may form in problems such as lid-driven cavity and dam-break. Artificial damping, used in the hydrostatic pressure example, considerably speeds up the calculation. Since the equation of state is used for fluids in MPM, the bulk modulus should be carefully selected to meet the compressibility requirement while reducing computation time. A reduced bulk modulus is used throughout the benchmark examples. According to our experience, when modeling water, a bulk modulus on the order of  $10^4$ - $10^6$  Pa seems to work for most cases. Pressure stabilization is effective for more complicated flow such as the lid-driven cavity, and the dam-break examples. The performance of B-spline is typically worse than that of GIMP especially near boundaries. However, it performs reasonably well for the lid-driven cavity and dam-break examples without the need of pressure stabilization.

It should be noted that the good results using GIMP are obtained with little changes in the original components of MPM. Like SPH, MPM-CFD is capable of simulating typical fluid dynamics problems with good results when compared to theory, as demonstrated in this paper. Fluid is treated similarly in SPH and MPM-CFD since both methods dis-

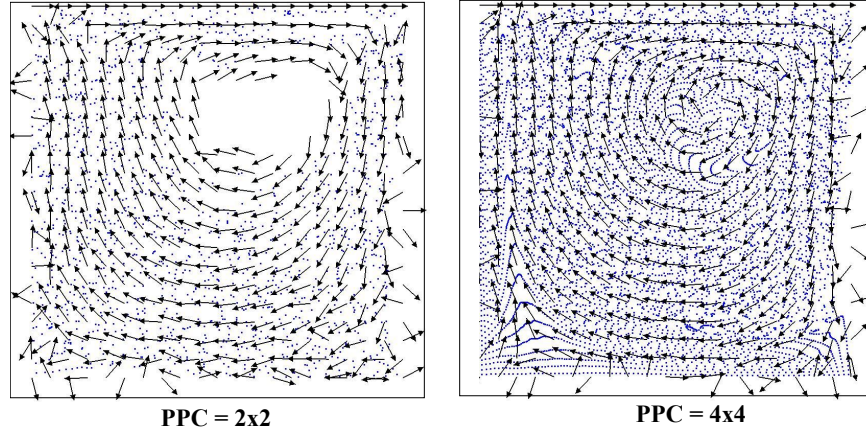


cretize the fluid domain using particles, and adopt similar constitutive models including the equation of state. For Poiseuille flows, Sigalotti et al. [2003] mentioned that SPH may encounter instability for  $Re=5.0$ , but not so for MPM-CFD. SPH and MPM-CFD share challenges in problems like the lid-driven cavity flow where particles tend to clump together, and voids can be produced due to large pressure fluctuations [Lee et al., 2008]. For SPH, the problem was mitigated via the adoption of a truly incompressible fluid algorithm that requires solving the pressure Poisson equation [Lee et al., 2008]. Although this algorithm has not been implemented in MPM, the use of pressure stabilization and a higher PPC in MPM does alleviate the issue and improves the results as demonstrated.

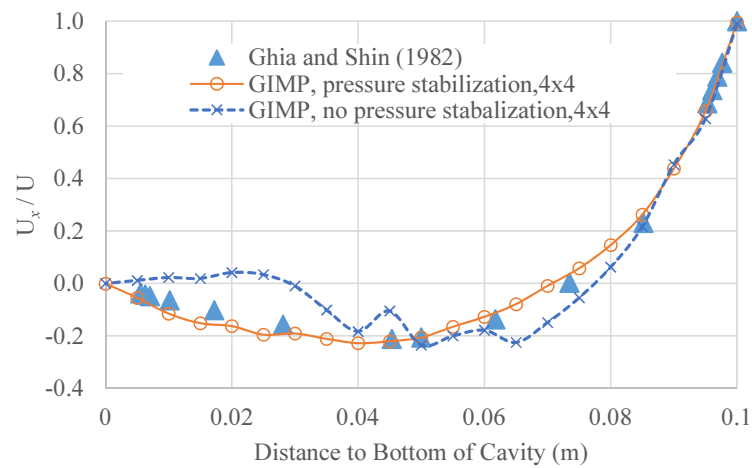
Our work shows that MPM-CFD is a promising computational method for fluid mechanics at low Reynolds numbers. Improved handling of incompressibility, and boundary conditions for the B-spline interpolation function could further enhance the capability of MPM for more complicated problems such as fluid-structure interaction.



(a) Problem setup of lid-driven cavity problem.  $Re = 100$ .



(b) Left: hole occurred for  $PPC=2 \times 2$ ; right: no hole for  $PPC=4 \times 4$ . Arrows indicate velocity at the grid nodes.  $t=0.2$  s.



(c) Comparison of  $U_x/U$  for GIMP results and experimental data. Pressure stabilization greatly improves the results.

Figure 2.5: GIMP simulation of the lid-driven cavity problem for  $Re = 100$ .

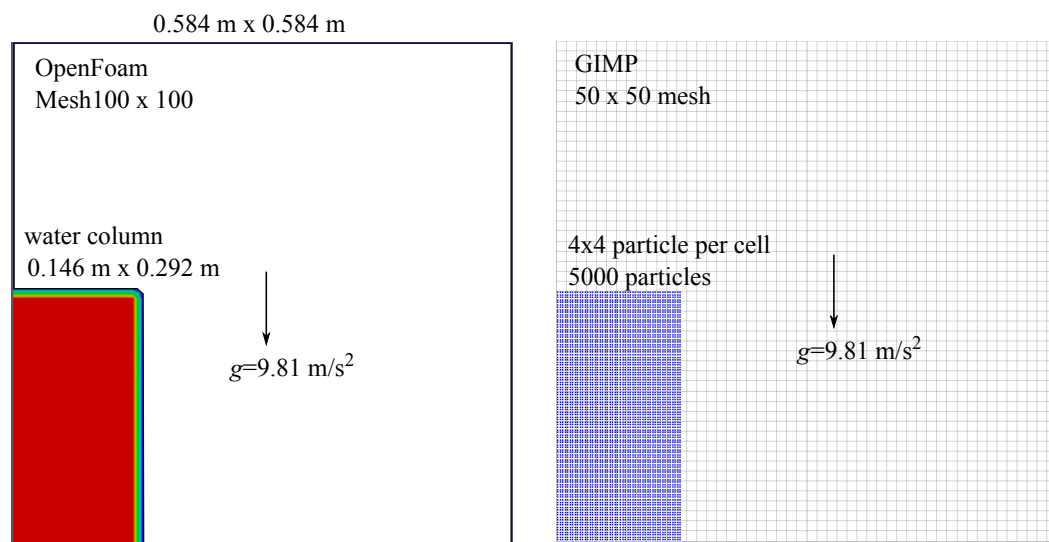
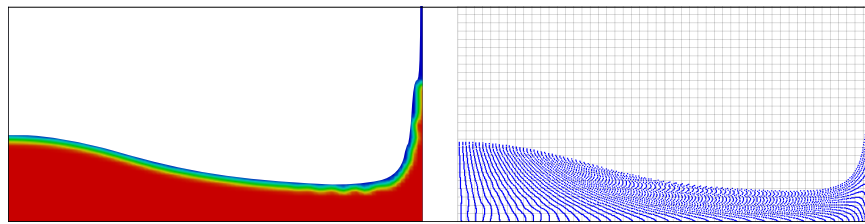
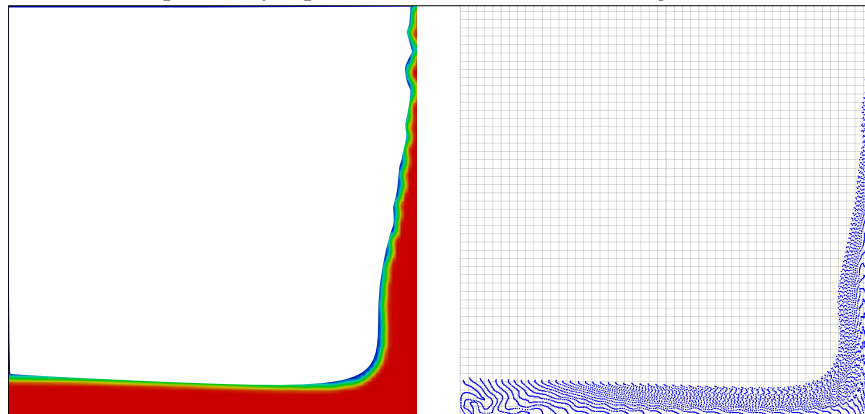


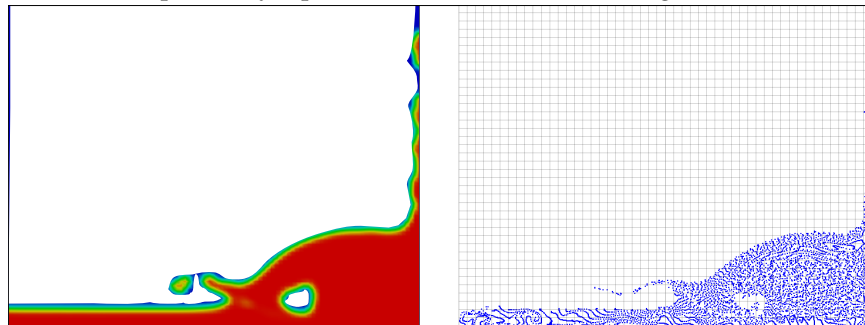
Figure 2.6: Setup of the dam-break problem for OpenFOAM and GIMP.



(a) Flow profile by OpenFOAM (left) and GIMP (right) at  $t=0.3$  s.

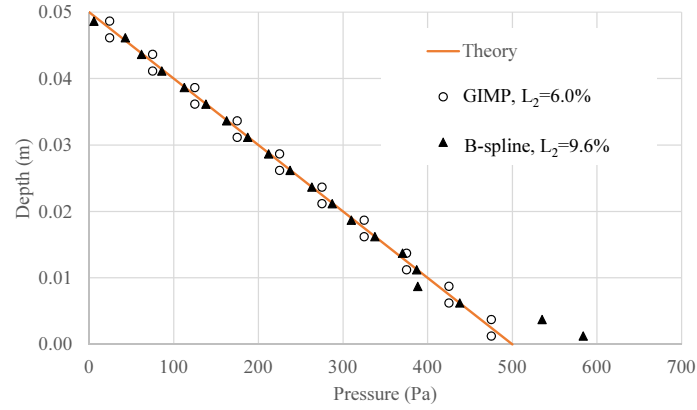


(b) Flow profile by OpenFOAM (left), and GIMP (right) at  $t=0.5$  s.

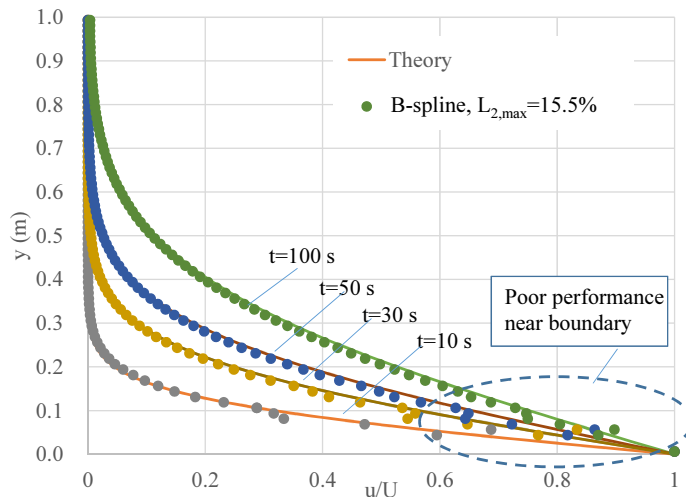


(c) Flow profile by OpenFOAM (left), and GIMP (right) at  $t=0.8$  s.

Figure 2.7: Simulation of dam-break using OpenFOAM and GIMP.



(a) Hydrostatic pressure.



(b) Rayleigh's problem.

Figure 2.8: B-spline results for: (a) hydrostatic pressure with comparison to GIMP and theory; (b) Rayleigh's problem with comparison to theory.

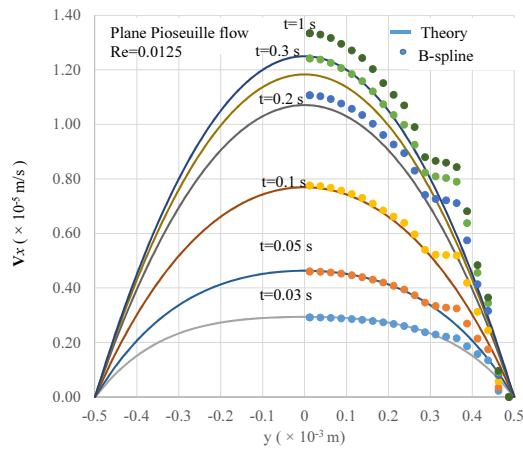
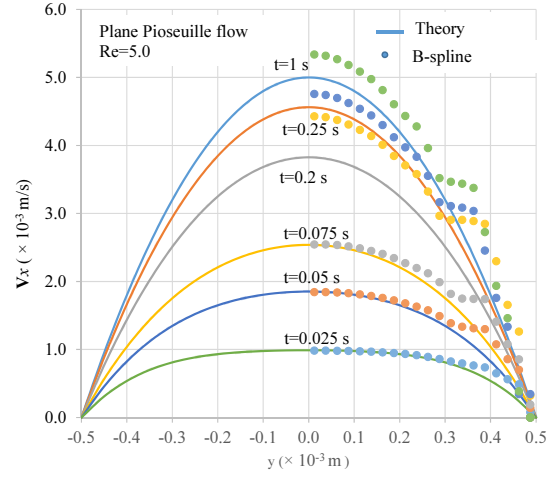
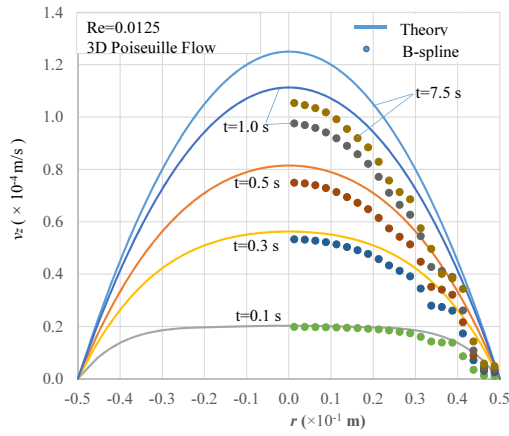
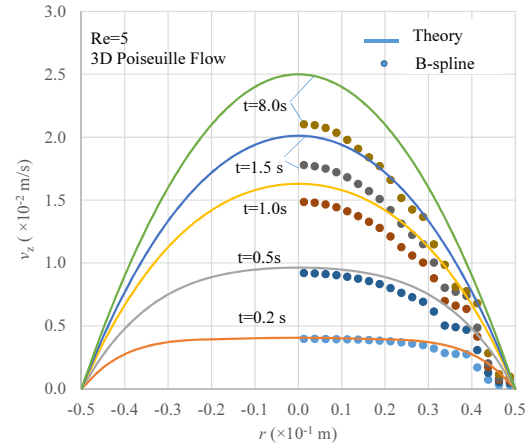
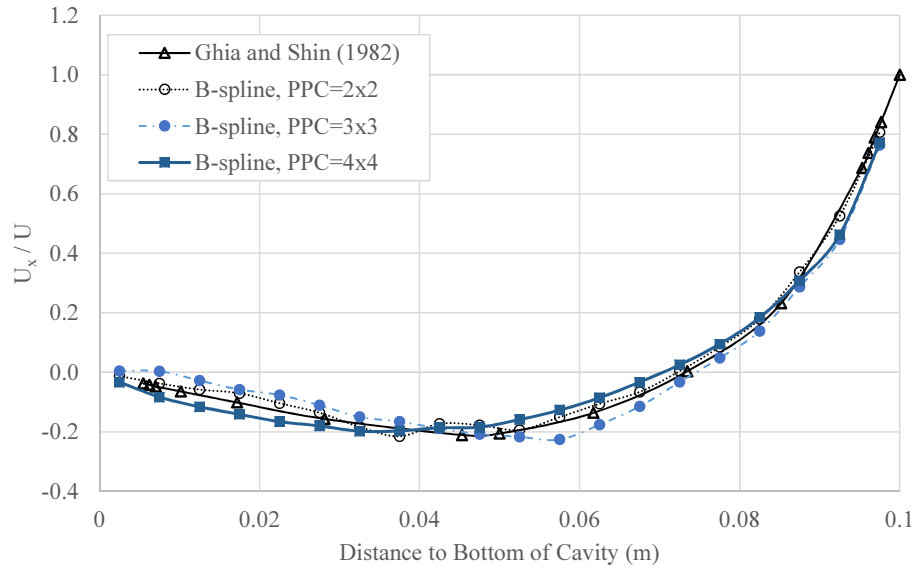
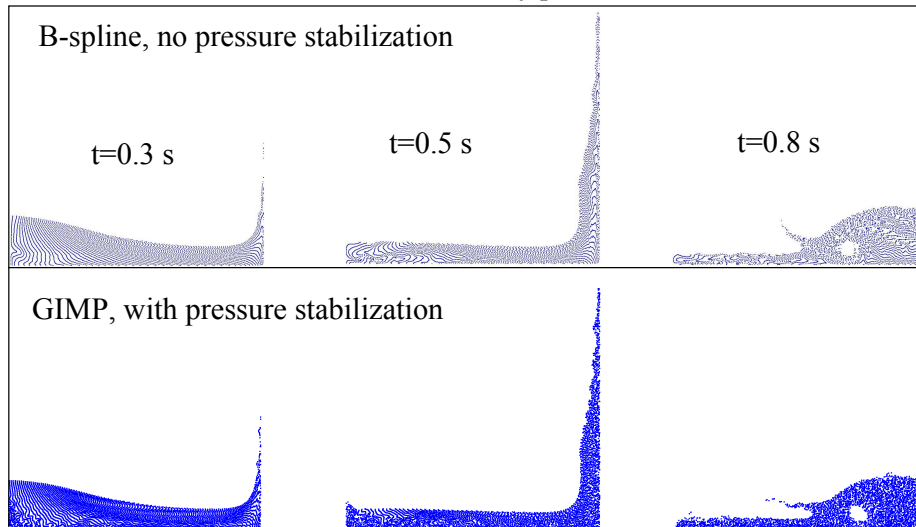
(a) Plane Poiseuille flow,  $Re=0.0125$ .(b) Plane Poiseuille flow,  $Re=5.0$ .(c) 3D Poiseuille flow,  $Re=0.0125$ .(d) 3D Poiseuille flow,  $Re=5$ .

Figure 2.9: B-spline results for 2D and 3D transient Poiseuille flows with comparison to theoretical results.



(a) Lid-driven cavity problem.



(b) Dam-break problem.

Figure 2.10: B-spline results without pressure stabilization for: (a) the lid-driven cavity problem as a function of PPC when compared with test data; (b) the dam-break problem when compared with results from GIMP with pressure stabilization.

## 2.6 References

- S. Bardenhagen and E. Kober. The generalized interpolation material point method. *CMES: Computer Modeling in Engineering and Sciences*, 5:477–495, 2004.
- J.U. Brackbill and H.M. Ruppel. FLIP: a method for adaptively zoned, particle-in-cell calculation of fluid flows in two dimensions. *Computer Physics Communications*, 95:314–343, 1986.
- H.S. Carslaw and J.C. Jaeger. *Conduction of Heat in Solids*. Oxford University Press, 1986.
- J.S. Chen, C. Pan, and T.Y.P. Chang. On the control of pressure oscillation in bilinear-displacement constant-pressure element. *Computer Methods in Applied Mechanics and Engineering*, 128:137–152, 1995.
- L. Chen, J. H. Lee, and C.-f. Chen. On the modeling of surface tension and its applications by the generalized interpolation material point method. *CMES: Computer Modeling in Engineering and Sciences*, 86:199–223, 2012.
- Z. Chen and R. Brannon. An evaluation of the material point method. Technical Report SAND2002-0482, Sandia Laboratory, 2002.
- L. Cueto-Felgueroso, G. Mosquera I. Colominas, F. Navarrina, and M. Casteleiro. On the Galerkin formulation of the smoothed particle hydrodynamics method. *International Journal for Numerical Methods in Engineering*, 60:1475–1512, 2004.
- U. Ghia, K.N. Ghia, and C.T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 48:378–411, 1982.
- J.E. Guilkey, T.B. Harman, and B. Banerjee. An Eulerian-Lagrangian approach for simulating explosions of energetic devices. *Computers and Structures*, 85:660–674, 2007.
- G. Hou, J. Wang, and A. Layton. Numerical methods for fluid–structure interaction – a review. *Communications in Computational Physics*, 12:337–377, 2012.
- E.S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence, and P. Stansby. Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method. *Journal of Computational Physics*, 227:8417–8436, 2008.



- J. H. Lee and J. E. Guilkey. Grain-scale modeling of saturated soils using the generalized interpolation material point method. In *Proceedings of the 16th International Conference of the International Society for Terrain-Vehicle Systems*, 2008.
- R.L. Lee, P.M. Gresho, and R.L. Sani. Smoothing techniques for certain primitive variable solutions of the Navier-Stokes equations. *International Journal for Numerical Methods in Engineering*, 14:1785–1804, 1979.
- S. Li and W. K. Liu. Meshfree and particle methods and their applications. *Applied Mechanics Review*, 55:1–34, 2002.
- S. Ma, X. Zhang, and X.M. Qiu. Comparison study of MPM and SPH in modeling hypervelocity impact problems. *International Journal of Impact Engineering*, 36:272–282, 2009.
- J.J. Monaghan. Simulating free surface flow with SPH. *Journal of Computational Physics*, 110:399–402, 1994.
- J.P. Morris, P.J. Fox, and Y. Zhu. Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics*, 136:214–226, 1997.
- A. Narin and J.E. Guilkey. Axisymmetric form of the generalized interpolation material point method. *International Journal for Numerical Methods in Engineering*, 0:1–24, 2013.
- V.P. Nguyen, T. Rabczuk, S. Bordas, and M. Duflot. Meshless methods: A review and computer implementation aspects. *Mathematics and Computers in Simulation*, 79(3):763–813, 2008.
- OpenFOAM. *Open source Field Operation And Manipulation*. The OpenFOAM Foundation, 2013.
- S. Parker, J. Guilkey, and T. Harman. A component-based parallel infrastructure for the simulation of fluid-structure interaction. *Engineering with Computers*, 22:277–292, 2006.
- V. Rivola. Comparative Study of the CFD codes Mistral and OpenFOAM. Technical report, R-Tech, Hotel d’ Entreprises - Parc Technologique Cap Delta, 2007.
- P.N. Shankar and M.D. Deshpande. Fluid mechanics in the driven cavity. *Annual Review of Fluid Mechanics*, 32:93–136, 2000.

- L.D.G. Sigalotti, J. Jaime, E. Sira, Y. Melean, and A. Hasmy. SPH simulations of time-dependent Poiseuille flow at low Reynolds numbers. *Journal of Computational Physics*, 191:622–638, 2003.
- R. Staroszczyk. Simulation of dam-break flow by a corrected smoothed particle hydrodynamics method. *Archives of Hydro-Engineering and Environmental Mechanics*, 57:61–79, 2010.
- M. Steffen, R.M. Kirby, and M. Berzins. Analysis and reduction of quadrature errors in the material point method (MPM). *International Journal for Numerical Methods in Engineering*, 76:922–948, 2008a.
- M. Steffen, P.C. Wallstedt, J.E. Guilkey, R.M. Kirby, and M. Berzins. Examination and analysis of implementation choices within the material point method (MPM). *CMES: Computer Modeling in Engineering and Sciences*, 31:107–127, 2008b.
- D. Sulsky, Z. Chen, and H.L. Schreyer. A particle method for history dependent materials. *Computer Methods in Applied Mechanics and Engineering*, 118:179–196, 1994.
- D. Sulsky, S.J. Zhou, and H.L. Schreyer. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications*, 87:236–252, 1995.
- P.C. Wallstedt and J.E. Guilkey. An evaluation of explicit time integration schemes for use with the generalized interpolation material point method. *Journal of Computational Physics*, 227:9628–9642, 2008.
- A. R. York, D. L. Sulsky, and H. L. Schreyer. The material point method for simulation of thin membranes. *International Journal for Numerical Methods in Engineering*, 44:1429–1456, 1999.



## Chapter 3

### On the Modeling of Surface Tension and its Applications by the Generalized Interpolation Material Point Method<sup>1</sup>

#### 3.1 Abstract

This paper presents a numerical procedure to model surface tension using the Generalized Interpolation Material Point (GIMP) method which employs a background mesh in solving the equations of motion. The force due to surface tension is formulated at the mesh grid points by using the continuum surface force (CSF) model and then added to the equations of motion at each grid point. In GIMP, we use the grid mass as the color function in CSF and apply a moving average smoothing scheme to the grid mass to improve the accuracy in calculating the surface interface. The algorithm, named as GIMP-CSF, is implemented using the software package Uintah and benchmarked by three numerical examples: static equilibrium of a 2D liquid drop, dynamic evolution of a square drop in 2D and 3D, and the capillary rise. The benchmark results, when compared to analytical solutions and those obtained by other approaches, demonstrated the accuracy and effectiveness of the GIMP-CSF algorithm.

#### 3.2 Introduction

Surface tension exists at the interface of two immiscible fluids or at the junctions of solid, fluid, and gas. As a traction force that is proportional to the magnitude of curvature at the interface, surface tension exists wherever the interface is not flat. In particular, at small scales, surface tension becomes comparable in magnitude to other forces that govern fluid motion due to the large surface-to-volume ratio at these scales. An effective numerical method capable of modeling surface tension in its interaction with changing boundaries can be very useful to solve surface tension related problems.

Surface tension has been modeled by the volume of fluid (VOF) method [Busmann et al., 1999, Gueyffier et al., 1999, Sussman, 2003], level set method [Sussman et al., 1994, Chang et al., 1996, Liu et al., 2000, Sussman, 2003], and the smoothed particle hydrodynamics (SPH) method [Nugent and Posch, 2000, Morris, 2000, Hu and Adams, 2009, Zhang, 2010]. These methods model surface tension approximately as a “volume force” based on the continuum surface force (CSF) approach [Brackbill et al., 1992].

---

<sup>1</sup>Published as L. Chen, J.H. Lee, and C.-f. Chen. On the modeling of surface tension and its applications by the generalized interpolation material point method. *CMES:Computer Modeling in Engineering and Sciences*, 86:199-223,2012.

Meshless or meshfree methods, such as SPH, have been gaining popularity due to their potential in dealing with complicated problems that involve failure, fragmentation and interactions of bodies with contact [Li and Liu, 2002]. The material point method (MPM) [Sulsky et al., 1994] and Generalized Interpolation Material Point (GIMP) method [Bardenhagen and Kober, 2004] are promising meshfree methods in solving solid mechanics problems with large deformations to address mesh tangling and other issues. In particular, the GIMP method was developed to improve the instability issue in the original material point method. MPM has been applied to many engineering problems including contact [Bardenhagen et al., 2001], crack and failure analysis [Nairn, 2003], explosion [Guilkey et al., 2007], and brittle failure of disc particles under impact [Li et al., 2011].

MPM [Sulsky et al., 1994] is an extension to solid mechanics from the fluid-implicit particle (FLIP) and particle in cell (PIC) methods in modeling highly distorted flow problems [Brackbill and Ruppel, 1986]. Consequently, MPM might be a suitable computational method to model fluid, structure and their interactions. However, to this date, the application of MPM for solving fluid-related problems has been limited [York et al., 1999, Lee and Guilkey, 2008, Gan et al., 2011]. In particular, the feasibility of modeling surface tension, as an important building block for the much more complicated problem of fluid-structure interactions at small scales, using MPM has not been demonstrated.

The objective of this paper is then to expand the modeling capability of the GIMP method to include surface tension. The paper is organized as follows. In Section 2 we review the GIMP method. The formulation of surface tension force by the CSF method for GIMP, and the numerical implementation of surface tension into an existing software package Uintah [Parker et al., 2006] are given in Section 3. The newly developed algorithm is verified in Section 4 by a few numerical examples and compared to analytical solutions. Finally, discussions and major conclusion are given in Section 5.

### 3.3 Review of the GIMP Method

The numerical procedure of GIMP and other MPM-related methods consists of three major steps. First, the objects of interest are discretized into particles (i.e., material points) which carry all the material properties and history-dependent state variables. Second, a computational grid is constructed to cover the anticipated domain of simulation. Particle information is then mapped to the grid points where the equations of motion are solved to update the acceleration, velocity, and displacement at the grid points. Third, the status

of each material point (such as displacement, velocity, acceleration, and state variables) is updated by mapping the information from the grid points. The procedure is repeated for each time step; details follow.

Using a circular disk (Fig. 3.1) for illustration, we review the GIMP method with a focus on the inclusion of the surface tension force in the equations of motion.

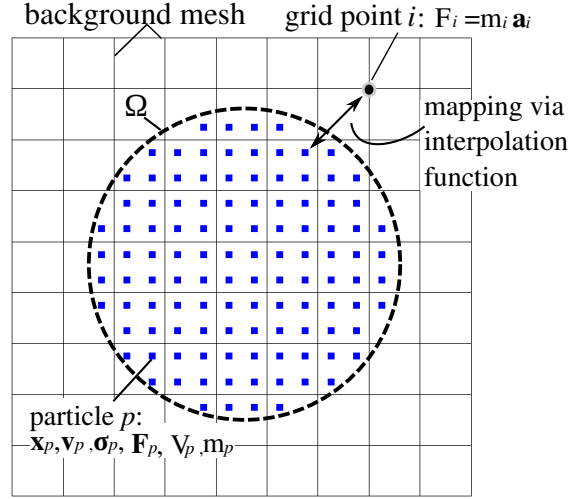


Figure 3.1: Illustration of the procedure of GIMP. At the beginning of computation, the disk  $\Omega$  is discretized into a collection of particles (i.e., material points) which are distributed over a background mesh. Each particle carries the local material properties and state variables of the disk: stress  $\sigma_p$ , mass  $m_p$ , volume  $V_p$ , deformation gradient  $\mathbf{F}_p$ , position  $\mathbf{x}_p$ , and velocity  $\mathbf{v}_p$ . An interpolation function is then selected to map, in each time increment, the information from the particles to the grid points such that acceleration at each grid point can be solved via the equations of motion  $\mathbf{F}_i = m_i \mathbf{a}_i$ . Information at the grid points is then mapped back to the particles to complete the calculation of one time increment.

At the beginning of computation, the disk which occupies the domain  $\Omega$  is discretized into  $N_p$  particles. A background mesh, also called the grid, is then constructed to cover the domain of simulation. The mesh consists of cells and grid points (nodes). The subscript  $p$  denotes a particle (i.e., a material point) and the subscript  $i$  indicates a node of the background mesh. In MPM, particles carry information including the material properties, mass ( $m_p$ ), volume ( $V_p$ ), all the state variables such as the stress tensor ( $\sigma_p$ ) and the deformation gradient ( $\mathbf{F}_p$ ), and kinematic variables such as the velocity ( $\mathbf{v}_p$ ). The particles may also carry external forces such as the body force  $\mathbf{b}_p$  and surface traction  $\tau_p$ .

At the beginning of each time step, the mass of the particle is mapped (extrapolated) to

node  $i$ :

$$m_i = \sum_p S_{ip}(\mathbf{x}_p) m_p \quad (3.1)$$

where  $m_i$  is the mass at the node,  $\sum_p$  denotes summation over particles, and  $S_{ip}$  is the interpolation function. In this paper, we used the GIMP interpolation function [Bardenhagen and Kober, 2004] for  $S_{ip}$  which is computationally more robust than those used in the conventional MPM. Additional discussions on the implications of the interpolation function  $S_{ip}$  can be found in Steffen et al. [2008].

The internal force at grid point  $i$ ,  $\mathbf{f}_i^{int}$ , is calculated from the particle stress  $\boldsymbol{\sigma}_p$  as:

$$\mathbf{f}_i^{int} = V_p \sum_p \mathbf{G}_{ip}(\mathbf{x}_p) \cdot \boldsymbol{\sigma}_p \quad (3.2)$$

where  $\mathbf{G}_{ip} = \nabla S_{ip}$  is the gradient of the interpolation function, and  $V_p$  is the discretized volume at particle  $p$ .

The external force at grid point  $i$ ,  $(\mathbf{f}_i^{ext})$ , includes contributions from the particle body force ( $\mathbf{b}_p$ ), and particle traction force ( $\boldsymbol{\tau}_p$ ):

$$\mathbf{f}_i^{ext} = \sum_p S_{ip}(\mathbf{x}_p) \mathbf{b}_p + \sum_p S_{ip}(\mathbf{x}_p) \boldsymbol{\tau}_p \quad (3.3)$$

At the grid, the equations of motion are:

$$\mathbf{m} \mathbf{a} = \mathbf{f}^{ext} - \mathbf{f}^{int} + \mathbf{f}^{sur} \quad (3.4)$$

where  $\mathbf{m}$  is the mass matrix,  $\mathbf{a}$  is the acceleration vector,  $\mathbf{f}^{ext}$  is the external force vector,  $\mathbf{f}^{int}$  is the internal force vector, and  $\mathbf{f}^{sur}$  represents the force contributed by surface tension when present.

Here, except for the surface tension force  $\mathbf{f}^{sur}$ , all the terms in Eq. 4.6 are already explicitly formulated. The acceleration at grid point  $i$  can be calculated by:

$$\mathbf{a}_i = (\mathbf{f}_i^{ext} - \mathbf{f}_i^{int} + \mathbf{f}_i^{sur}) / m_i \quad (3.5)$$

The formulation of the surface tension force  $\mathbf{f}_i^{sur}$  will be presented shortly in Section 3.4.2. Once the grid acceleration is solved, the grid velocity  $\mathbf{v}_i$  can be calculated via conservation

of momentum:

$$\mathbf{v}_i = \frac{\sum_p S_{ip}(\mathbf{x}_p) \mathbf{v}_p m_p}{m_i} \quad (3.6)$$

After the grid velocity is calculated, one needs to update the state of each particle using  $\mathbf{a}_i$  and  $\mathbf{v}_i$ . For a given time step  $\Delta t$ , the increment of the particle velocity,  $\Delta \mathbf{v}_p$ , is calculated from:

$$\Delta \mathbf{v}_p = \Delta t \sum_i S_{ip}(\mathbf{x}_i) \mathbf{a}_i \quad (3.7)$$

where  $\sum_i$  denotes summation over the grid points. It should be noted that the same interpolation function is used for the mapping from grid to particles as well as from particles to grid. The increment in the particle position,  $\Delta \mathbf{x}_p$ , is updated from  $\mathbf{v}_i$  and  $\mathbf{a}_i$ :

$$\mathbf{v}_i^L = \mathbf{v}_i + \mathbf{a}_i \Delta t \quad (3.8)$$

$$\Delta \mathbf{x}_p = \sum_i S_{ip} \mathbf{v}_i^L \quad (3.9)$$

where  $\mathbf{v}_i^L$  is the updated grid velocity. Particle position and velocity are then updated using Eq. 4.12 and Eq. 4.13.

The increment in the stress at particles,  $\Delta \sigma_p$ , is updated using the update-stress-last (USL) algorithm [Wallstedt and Guilkey, 2008]:

$$\Delta \mathbf{F}_p = \mathbf{I} + \Delta t \sum_i \mathbf{G}_{ip} \mathbf{v}_i^L \quad (3.10)$$

$$\Delta \sigma = f(\Delta \mathbf{F}_p) \quad (3.11)$$

where  $\Delta \mathbf{F}_p$  is the increment of the deformation tensor,  $\mathbf{I}$  is the identity tensor, and  $f(\Delta \mathbf{F}_p)$  is a function associated with the constitutive law of the material.

For nearly-incompressible fluids of interest in this paper, we adopt the constitutive law given in Monaghan et al. [1999], Cueto-Felgueroso et al. [2004] to calculate the stress:

$$\sigma = -p \mathbf{I} + 2\mu \mathbf{d}' \quad (3.12)$$

where  $\mu$  is the dynamic viscosity,  $\mathbf{d}'$  is the rate of deformation tensor based on  $\Delta \mathbf{F}_p$ , and  $p$



is the hydrostatic pressure determined by an equation of state:

$$p = K \left[ \left( \frac{\rho}{\rho_0} \right)^\gamma - 1 \right] \quad (3.13)$$

where  $K$  is the bulk modulus,  $\rho$  the fluid density,  $\rho_0$  the initial density, and  $\gamma$  a constant. This completes the computation of a typical time step.

### 3.4 Modeling of Surface Tension

Expression of the surface tension force  $\mathbf{f}_i^{sur}$  in Eq. 3.5 is elaborated in this section. We adopt the CSF model [Brackbill et al., 1992] for the surface tension force since it is capable of handling complicated immiscible interfaces in the presence of contact angle. In the following, the CSF method is first reviewed, an approach is then proposed to include surface tension in GIMP.

#### 3.4.1 Review of the CSF method

Fig. 3.2 illustrates how the CSF method models the surface tension between two immiscible

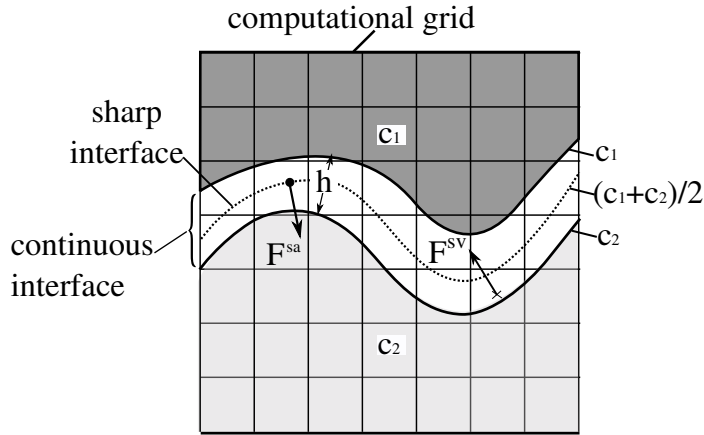


Figure 3.2: The CSF method for fluids with different colors ( $c_1$  and  $c_2$ ). Over the computational grid, contours of the continuous color function  $c(\mathbf{x})$  are illustrated showing also a transition zone of width  $h$ . Surface tension force,  $\mathbf{F}^{sa}$ , is reformulated as a volume force  $\mathbf{F}^{sv}$  within the transition zone.

fluids. The fluids are represented by different color functions ( $c_1$  and  $c_2$ ) (e.g., 0 and 1). Although color functions should be discontinuous at the interface in reality, a transition zone is used in the CSF method to bridge the discontinuity between  $c_1$  and  $c_2$  using the color

function  $c(\mathbf{x})$  over the transition zone (i.e., the continuous interface labeled in Fig. 3.2).  $c(\mathbf{x})$  will converge to a step function as the width of the transition zone,  $h$ , approaches zero.

In the CSF method, the surface tension is modeled approximately as a body force or volume force, as opposed to its physical meaning by which the surface tension is force per length. The volume force, denoted by  $\mathbf{F}^{sv}$ , acts within the transition zone, shown in Fig. 3.2, in a continuous manner [Brackbill et al., 1992] :

$$\mathbf{F}^{sv}(\mathbf{x}) = \sigma \kappa(\mathbf{x}) \frac{\nabla c(\mathbf{x})}{[c]} \quad (3.14)$$

where  $\sigma$  is the coefficient of surface tension,  $\nabla c(\mathbf{x})$  is the gradient of the color function, and  $[c]$  is the jump of color function across the fluid interface ( $[c] = |c_1 - c_2|$ ).  $\kappa$  represents the curvature evaluated at  $\mathbf{x}$  at the interface; in computation, however, this curvature term is approximated by the gradient of the unit vector  $\mathbf{n}$  at  $\mathbf{x}$ :

$$\kappa(\mathbf{x}) = \nabla \cdot \mathbf{n}(\mathbf{x}) = \nabla \cdot \left( \frac{\nabla c(\mathbf{x})}{|\nabla c(\mathbf{x})|} \right) \quad (3.15)$$

It should be noted that, in order to better approximate the curvature at the transition zone, the color function  $c(\mathbf{x})$  can be further improved using a smoothed color function  $\tilde{c}(\mathbf{x})$  [Brackbill et al., 1992, Sussman, 2003]:

$$\tilde{c}(\mathbf{x}) = \psi * c(\mathbf{x}) = \int_{\Omega_\psi} c(\mathbf{x}') \psi(\mathbf{x}' - \mathbf{x}) d\mathbf{x}' \quad (3.16)$$

where  $\Omega_\psi$  is the smoothing range of the smoothing kernel  $\psi$ . Various smoothing kernels have been developed such as the Nordmark kernel [Nordmark, 1991, Williams et al., 1999], the B-spline kernel [Boor, 1967, Brackbill et al., 1992], and the quintic spline kernel used in SPH [Morris et al., 1997, Hu and Adams, 2009]. In our study, a moving average smoothing kernel is used to be discussed in Section 3.4.4.

### 3.4.2 Formulation of surface tension in GIMP

We include the force attributed to surface tension as an additional external force to the equations of motion in Eq. 4.6. The modeling of surface tension is based on Eq. 3.14 together with a choice of the color function. Here we adopt the grid mass,  $m_i$ , as the color function  $c(\mathbf{x})$ . Therefore, the volume force due to surface tension at grid point  $i$  can be

formulated as:

$$\mathbf{F}_i^{sv} = \sigma \kappa_i \frac{\nabla m_i}{[m_i]} \quad (3.17)$$

where each term is formulated as follows. First,  $\nabla m_i$ , the gradient of grid mass, is calculated by:

$$\nabla m_i = \sum_{i'} \mathbf{G}_{ip}(\mathbf{x}_{i'}) m_{i'} \quad (3.18)$$

where  $\sum_{i'}$  is a summation over the neighboring nodes at  $\mathbf{x}_{i'}$ .

Second,  $[m_i]$  is defined as  $(\rho_1 - \rho_2)V_c$ , where  $\rho_1$  and  $\rho_2$  are the densities of the two immiscible fluids, respectively, and  $V_c$  is the volume of the cell of the background mesh.

Third, the curvature  $\kappa_i$  is calculated by:

$$\kappa_i = \nabla \cdot \mathbf{n}_i = \sum_{i'} \mathbf{G}_{ip}(\mathbf{x}_{i'}) \cdot (\mathbf{n}_{i'})^T \quad (3.19)$$

where  $T$  denotes the transpose of a vector. In turn, the unit normal vector used in the curvature calculation is based on a smoothed grid mass  $\tilde{m}_i$ :

$$\mathbf{n}_i = \left( \frac{\nabla \tilde{m}_i}{|\nabla \tilde{m}_i|} \right) \quad (3.20)$$

where  $\nabla \tilde{m}_i$  is evaluated using Eq. 3.18. In Section 3.4.4 we discuss the application of moving average to smooth the mass at each grid point.

Once the volume force is calculated, the force contributed by surface tension,  $\mathbf{f}^{sur}$ , in the equations of motion in Eq. 4.6 can be calculated by:

$$\mathbf{f}^{sur} = V_c \mathbf{F}_i^{sv} \quad (3.21)$$

It should be noted that Eq. 3.17 can be further improved [Brackbill et al., 1992] by:

$$\mathbf{F}_i^{sv} = \sigma \kappa_i \frac{\nabla m_i}{[m_i]} \frac{m_i}{< m_i >} \quad (3.22)$$

where  $< m_i > = V_c(\rho_1 + \rho_2)/2$ . Eq. 3.22 provides additional computational advantage in calculating the acceleration using GIMP:

$$\mathbf{a}_i^{sur} = \frac{V_c \mathbf{F}_i^{sv}}{m_i} = \sigma \kappa_i \frac{\nabla m_i}{(\rho_2 - \rho_1) < m >} \quad (3.23)$$

where  $\mathbf{a}_i^{sur}$ , the acceleration at grid point  $i$  due to surface tension, depends solely on the gradient of grid mass.

### 3.4.3 Treatment of contact angle as a boundary condition

In wetting problems, the balance in surface tension among solid, liquid, and gas results in the formation of the contact angle whose orientation is defined by the tangent of the surface profile at the point where the three phases meet. To expand GIMP's capability of handling the three-phase problem, we model the contact angle as a boundary condition illustrated in Fig. 3.3.

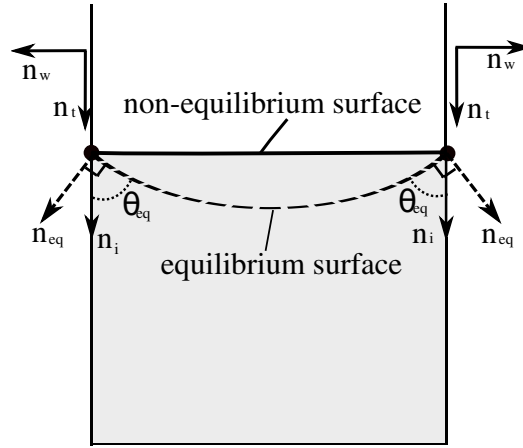


Figure 3.3: Contact angle  $\theta_{eq}$  as a boundary condition.  $\mathbf{n}_{eq}$  is the unit vector normal to the tangent along  $\theta_{eq}$ ;  $\mathbf{n}_t$  is a unit vector tangent to the wall;  $\mathbf{n}_w$  is a unit vector normal to the wall; and  $\mathbf{n}_i$  is the unit normal vector along the liquid-wall contact line. During simulation, the profile of the liquid in equilibrium can be determined when  $\mathbf{n}_i$  is aligned with  $\mathbf{n}_{eq}$ .

At the boundary where the wall interacts with two fluids (i.e., the liquid and gas), the unit normal calculated earlier in Eq. 3.20 is replaced by  $\mathbf{n}_{eq}$ :

$$\mathbf{n}_{eq} = \mathbf{n}_w \cos \theta_{eq} + \mathbf{n}_t \sin \theta_{eq} \quad (3.24)$$

where  $\mathbf{n}_w$  and  $\mathbf{n}_t$  are normal and tangent to the wall, respectively (Fig. 3.3). The replacement of  $\mathbf{n}_i$  by  $\mathbf{n}_{eq}$  in Eq. 3.20 for all the grid points at the boundary (where the meniscus is) will force the liquid profile to comply with a given contact angle. Consequently, the curvature of the liquid profile at the boundary will be updated incrementally during simulation, eventually reaching equilibrium when  $\mathbf{n}_i$  aligns with  $\mathbf{n}_{eq}$ .

### 3.4.4 Implementation of GIMP-CSF

Fig. 3.4 outlines the main procedures discussed previously for our implementation of sur-

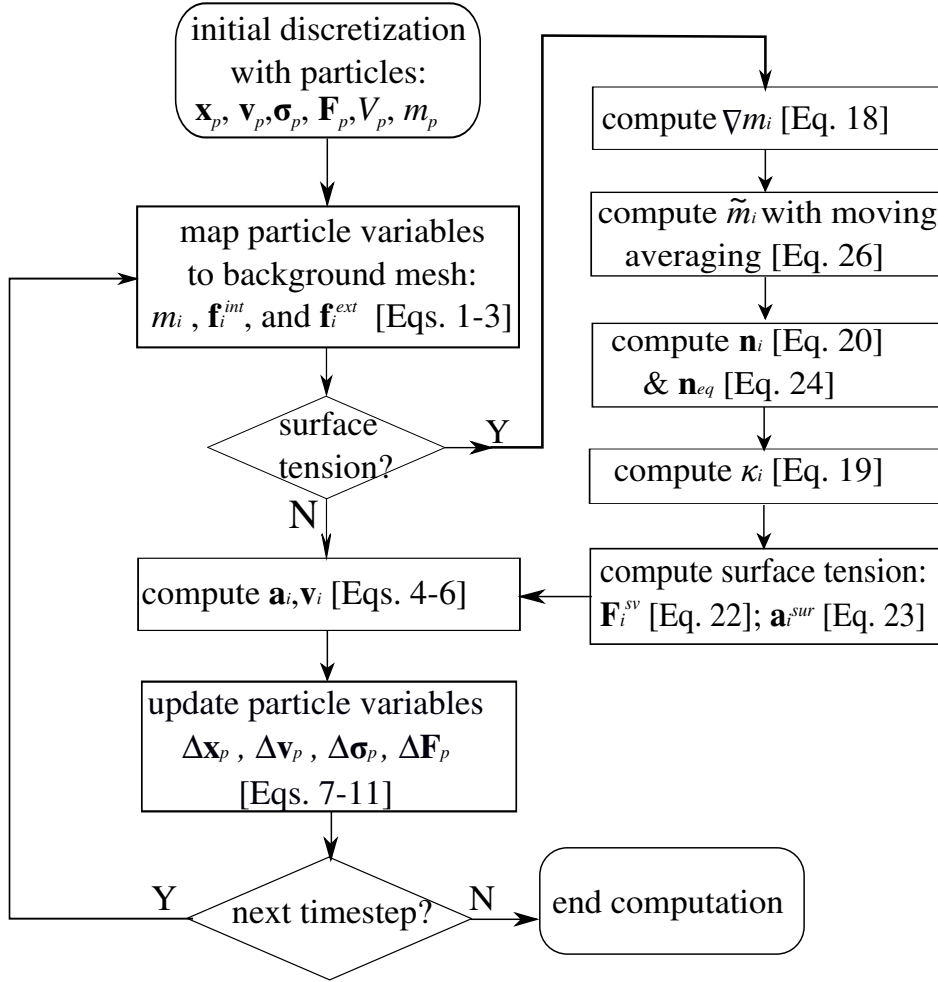


Figure 3.4: A flow chart for GIMP-CSF.  $\nabla m_i$ —gradient of grid mass;  $\tilde{m}_i$ —smoothed grid mass;  $\mathbf{n}_i$ —grid surface normal;  $\mathbf{n}_{eq}$ —surface normal based on contact angle;  $\kappa_i$ —grid curvature;  $\mathbf{F}_i^{sv}$ —grid surface tension (volume) force;  $\mathbf{a}_i^{sur}$ —grid surface tension (acceleration) force.

face tension in MPM. The algorithm is named GIMP-CSF since it integrates the CSF model of surface tension with the GIMP method. GIMP-CSF is implemented in Uintah [Parker et al., 2006], a comprehensive software package that includes the GIMP method. Our contribution to the improvement of Uintah is threefold: calculation of the gradient of grid mass, smoothing of the grid mass, and evaluation of the surface curvature; each aspect is presented below.

For illustration, we consider a 3D domain in Cartesian space meshed uniformly into  $L \times M \times N$  cells in the  $x$ ,  $y$ , and  $z$  direction, respectively. Each cell has volume  $V_c = \Delta x \Delta y \Delta z$ , where  $\Delta x$ ,  $\Delta y$ , and  $\Delta z$  are the cell sizes. The position of a grid node is represented by  $\mathbf{x}(i, j, k) = (i\Delta x, j\Delta y, k\Delta z)$ , where  $1 \leq i \leq L$ ,  $1 \leq j \leq M$ , and  $1 \leq k \leq N$ .

The gradient of the grid mass for a grid node at  $\mathbf{x}_{i,j,k}$  can be calculated by:

$$\nabla m_{i,j,k} = \sum_{i',j',k'} \mathbf{G}_{ip}(\mathbf{x}_{i',j',k'}) m_{i',j',k'} \quad (3.25)$$

where  $m_{i',j',k'}$  is the grid mass at the neighboring node  $\mathbf{x}_{i',j',k'}$ . With the GIMP interpolation function, there are 27 neighboring nodes where the gradient of interpolation function  $\mathbf{G}_{ip}$  is evaluated at each node.

Smoothing of the grid mass is conducted by using moving average as the smoothing kernel (cf. Eq. 3.16):

$$\tilde{m}_{i,j,k} = \frac{1}{n^3} \sum m_{i',j',k'} \quad (3.26)$$

where  $n$  is the number of neighboring nodes in each coordinate direction; the summation is over neighboring nodes;  $\tilde{m}_{i,j,k}^x$ , and  $\tilde{m}_{i,j,k}^y$ , and  $\tilde{m}_{i,j,k}^z$  are the smoothed grid mass along each direction. We did not apply the smoothing average kernel at or near the boundaries since there are not sufficient neighboring nodes for Eq. 3.26. Further smoothing can be done by conducting more iterations of the smoothing process by substituting the left-hand side of Eq. 3.26 to its right-hand side.

With  $\tilde{m}_{i,j,k}$  found, the unit normals are obtained by:

$$\mathbf{n}_{i,j,k} = \frac{\nabla \tilde{m}_{i,j,k}}{|\nabla \tilde{m}_{i,j,k}|} \quad (3.27)$$

where  $\nabla \tilde{m}$  is calculated similar to  $\nabla m$  in Eq. 3.25. It should be noted that if contact angle is specified, the boundary conditions described in Section 3.4.3 will be applied to modify  $\mathbf{n}_{i,j,k}$ .

The curvature is evaluated via:

$$\begin{aligned} \kappa_{i,j,k} &= \nabla \cdot \mathbf{n}_{i,j,k} \\ &= \sum_{i',j',k'} G_{ip} \cdot \mathbf{n}_{i',j',k'}^T \end{aligned} \quad (3.28)$$

### 3.5 Numerical Examples

Three numerical examples are given in this section to assess the efficacy of the GIMP-CSF algorithm implemented in Uintah. These examples consider only single-phase fluids with surface tension, applicable to situations when the effect of one phase on the other is negligible [Blanchette and Bigioni, 2009, Zhang, 2010]. Although the algorithm is also capable of simulating surface tension problems in multi-phase fluids, they are beyond the scope of this paper.

We used the fluid properties listed in Tab. 3.1 and GIMP interpolation function given in Bardenhagen and Kober [2004] for all the examples below.

Table 3.1: Fluid properties used in the numerical examples

Surface tension (dynes/cm)	Density (g/cm <sup>3</sup> )	Bulk modulus ( dynes/cm <sup>2</sup> )	Dynamic viscosity (dynes · s/cm <sup>2</sup> )	$\gamma$
2.4	1	$1.5 \times 10^5$	0.5	7.0

#### 3.5.1 2D equilibrium droplet

A droplet in equilibrium experiences a pressure drop,  $p_{drop}$ , across the interface due to surface tension. The theoretical solutions for  $p_{drop}$  are

$$p_{drop} = \sigma \kappa = \sigma/R, \text{ for 2D} \quad (3.29)$$

$$p_{drop} = 2\sigma \kappa = 2\sigma/R, \text{ for 3D} \quad (3.30)$$

where  $\sigma$  is the surface tension coefficient, and  $R$  the radius of the droplet. In this example, we simulate the 2D case with GIMP-CSF and compare the results with those obtained by a different approach [Brackbill et al., 1992].

A 1-cm-radius droplet in the  $x$ - $y$  plane is simulated with a typical set-up shown in Fig. 3.5. The droplet has an initial non-circular shape and is discretized into 1264 particles; the initial shape was chosen to be close to a circle to save computational time. The problem domain is overlaid with a  $4 \times 4$  cm grid with  $40 \times 40$  cells; there are 2 particles per cell in the  $x$  and  $y$  directions, respectively (PPC= $2 \times 2$ ). In the  $z$  direction, there is only one layer of particles and cells for this 2D case. Besides the set-up in Fig. 3.5, various numbers of PPC and cells were also investigated.

As an example, Fig. 3.6 shows the contours of grid mass and its gradient ( $\nabla m$ ) for the

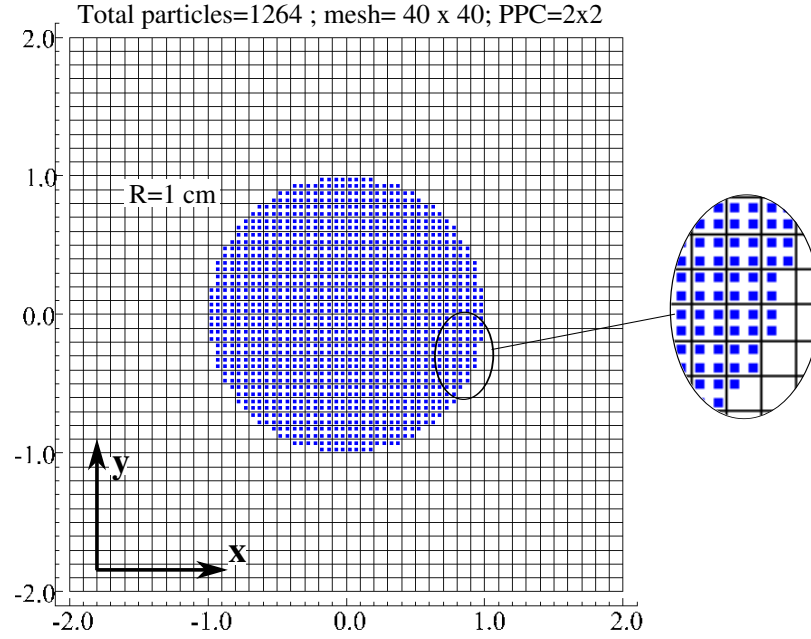


Figure 3.5: Example set-up of 2D equilibrium drop with GIMP using Uintah.

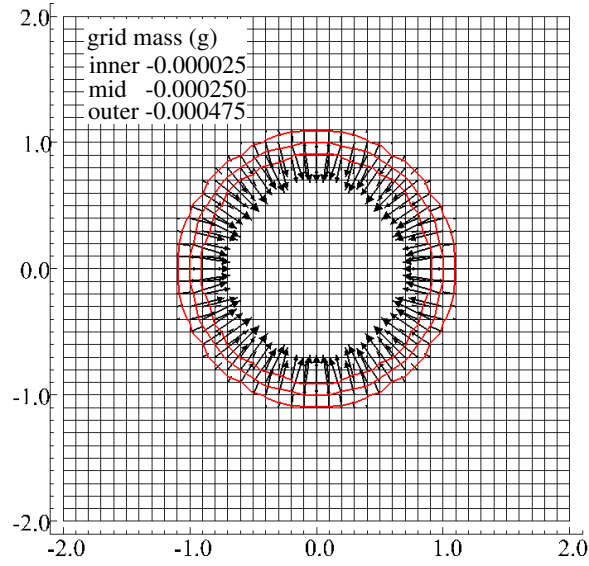


Figure 3.6: Contours of the grid mass and the gradients at t=0 s.

situation in Fig. 3.5 at t=0 s. The grid mass is mapped from the particle mass by Eq. 4.7, and its gradient calculated by Eq. 3.18. The contours of the grid mass define a transition zone which is about two cells wide. Only within the transition zone are the gradients nonzero, so is the surface tension force. Thus there is no need for the CSF model to track



the evolution of the boundary. Since the surface tension force is zero when  $\nabla m$  is zero, it can save computational time by skipping the calculation of the curvature and surface tension when the gradient  $\nabla m$  is smaller than a threshold (e.g., 10 % of  $[m]/\Delta x$ ). Fig. 3.7 shows the curvatures evaluated at the nodes when the gradient  $\nabla m$  is greater than the pre-

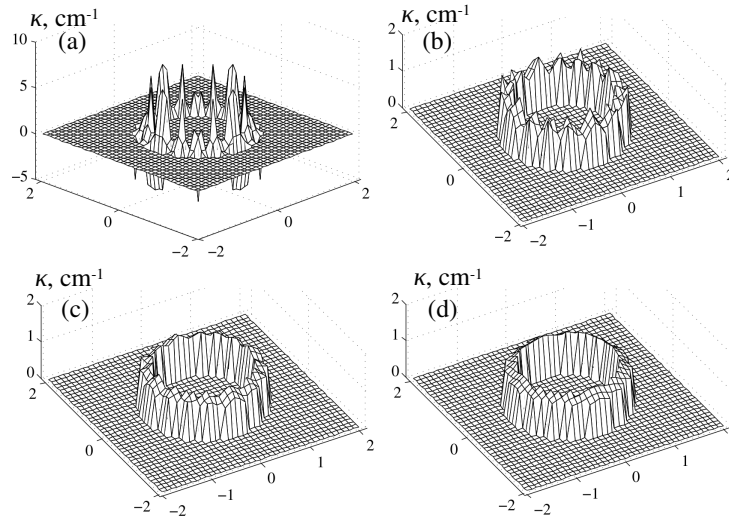


Figure 3.7: Curvatures ( $t=0$ ) with different smoothed grid mass: (a) zero iteration of moving averaging; (b) 1 iteration of moving averaging ; (c) 2 iterations of moving averaging; (d) 4 iterations of moving averaging. Theoretical curvature is  $1 \text{ cm}^{-1}$ .

defined threshold. It shows the influence of different smoothing schemes on the curvature. Without smoothing, Fig. 3.7a shows that the curvature has a large error compared to the theoretical value of  $1 \text{ cm}^{-1}$  at the boundary of the droplet. With 1 iteration of moving average smoothing, the curvature (Fig. 3.7b) is improved. More iterations of moving average smoothing can greatly improve the accuracy of the curvature, as shown in Fig. 3.7c and Fig. 3.7d. Similar results were also found in [Brackbill et al., 1992].

Next we plot the surface tension in terms of acceleration  $\mathbf{a}^{surf}$  (cf. Eq. 3.23) in Fig. 3.8, comparing the result of 1 iteration of the moving average smoothing with that of 4. The directions of surface tension at the grid points in Fig. 3.8a and Fig. 3.8b are the same since they are based on the gradient of unsmoothed grid mass (Fig. 3.6). However, the magnitude of the surface tension for each case is different since it is determined by the curvature which, in turn, is affected by the smoothing scheme used.

In a dynamic simulation, the droplet oscillates and will reach an equilibrium state. Given the initial configuration, we found that 0.25 second is sufficient for the droplet to sta-

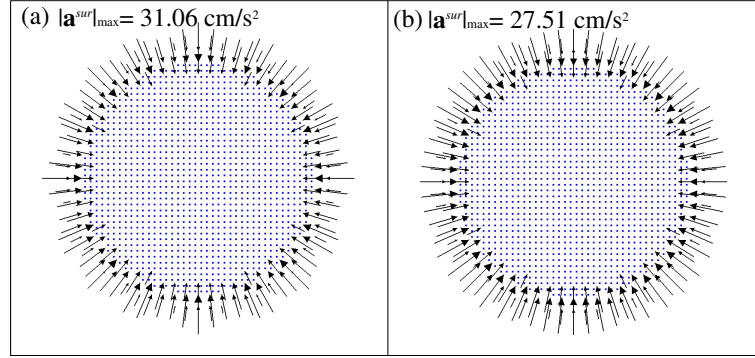


Figure 3.8: Surface tension,  $\mathbf{a}^{sur}$ , at  $t=0$  with different curvatures from (a) one iteration of moving averaging; (b) 4 iterations of moving averaging.

bilize. The grid pressure is then obtained by mapping the particle stress to the background mesh. Fig. 3.9 shows the distribution of pressure for different numbers of smoothing it-

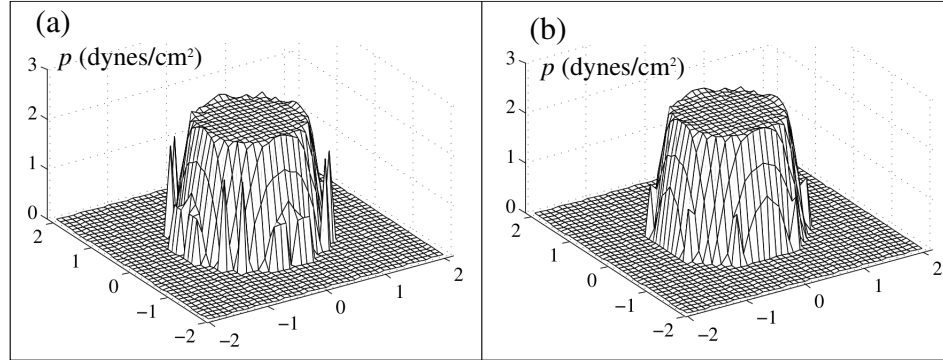


Figure 3.9: Pressure at  $t=2.5$  s from different smoothing schemes: (a) one iteration of moving averaging; (b) 4 iterations of moving averaging. Theoretical pressure drop is  $2.4 \text{ dynes/cm}^2$ .

erations. Accuracy of the calculated pressure can be assessed by the following measures [Brackbill et al., 1992]:

$$\langle p \rangle = \frac{1}{N_d} \sum_{i,j=1}^{N_d} p_{i,j} \quad (3.31)$$

$$L_2 = \left[ \frac{\sum_{i,j=1}^{N_d} (p_{i,j} - p_{drop})^2}{N_d p_{drop}^2} \right] \quad (3.32)$$

where  $\langle p \rangle$  is the average pressure,  $N_d$  is the number of grid nodes within the droplet radius,  $i$  and  $j$  are node indices, and  $L_2$  is the root-mean-square (rms) error relative to the theoretical pressure drop  $p_{drop}$  which is  $2.4 \text{ dyne/cm}^2$  for our case (Eq. 3.29). With Eq. 3.31

and Eq. 3.32, we have, for Fig. 3.9a,  $\langle p \rangle / p_{drop} = 98.36\%$  and  $L_2 = 0.043$ ; for Fig. 3.9b, we have  $\langle p \rangle / p_{drop} = 98.16\%$  and  $L_2 = 0.043$ . It shows that the results in pressure in Fig. 3.9 are very close such that the number of smoothing iterations has little effect on the distribution of pressure.

Tab. 3.2 shows the effects of grid resolution, number of particles per cell, and smoothing schemes. Generally, the GIMP-CSF method with smoothing has the same order of accuracy (less 5% error in  $\langle p \rangle$ ) as those in Brackbill et al. [1992]. Smoothing is shown to be very important; when not used, the error could be as high as 125.7%. However, smoothing once is adequate for this example as smoothing more times does not change the results significantly.

Table 3.2: Comparison of results using GIMP-CSF to those using the arbitrary Lagrangian-Eulerian (ALE) method for simulating a 2D equilibrium droplet. ( $\Delta h$  is the mesh size; PPC–particles per cell.)

ALE [Brackbill et al., 1992]				
$R/\Delta h$	# of particles (PPC)	smoothing	$\langle p \rangle / p_{drop}$	rms error (Eq. 3.32)
10	/	B-spline	1.034	0.056
20	/	B-spline	1.016	0.028
GIMP-CSF				
10	1264 (2×2)	none	1.641	0.655
10	1264 (2×2)	1 moving averaging	0.983	0.043
10	1264 (2×2)	2 moving averaging	0.982	0.042
10	1264 (2×2)	4 moving averaging	0.984	0.043
10	5024 (4×4)	none	1.212	0.279
10	5024 (4×4)	1 moving averaging	0.984	0.046
10	5024 (4×4)	2 moving averaging	0.992	0.045
10	5024 (4×4)	4 moving averaging	0.991	0.045
20	5024 (2×2)	none	2.257	1.286
20	5024 (2×2)	1 moving averaging	0.967	0.040
20	5024 (2×2)	2 moving averaging	0.980	0.031
20	5024 (2×2)	4 moving averaging	0.978	0.033
20	20108 (4×4)	none	1.941	0.971
20	20108 (4×4)	1 moving averaging	0.973	0.035
20	20108 (4×4)	2 moving averaging	0.989	0.026
20	20108 (4×4)	4 moving averaging	0.987	0.027

In terms of the effect of grid resolution, results in Tab. 3.2 indicate that a cell size ( $\Delta h$ ) 10% of the droplet radius ( $R/\Delta h = 10$ ) is adequate since the results are comparable to those with  $R/\Delta h = 20$ . Regarding the effect of the number of particles per cell, PPC=4×4 gives

slightly better results than  $\text{PPC}=2\times 2$ . For computational efficiency, we recommend a  $\text{PPC}=2\times 2$  (or  $2\times 2\times 2$  for 3D).

### 3.5.2 Evolution of non-equilibrium drops in 2D and 3D

In this example, we consider the evolution of a droplet from a square in 2D (or a cubic in 3D) to the circular (spherical) profile at equilibrium under surface tension. This example has been studied by other numerical methods [Brackbill et al., 1992, Zhang, 2010] and serves as another benchmark problem for our GIMP-CSF algorithm.

We first consider the 2D case where a square drop ( $1 \times 1$  cm) is simulated shown in Fig. 3.10. The drop is discretized with 400 particles and overlaid with a  $2 \times 2$  cm back-

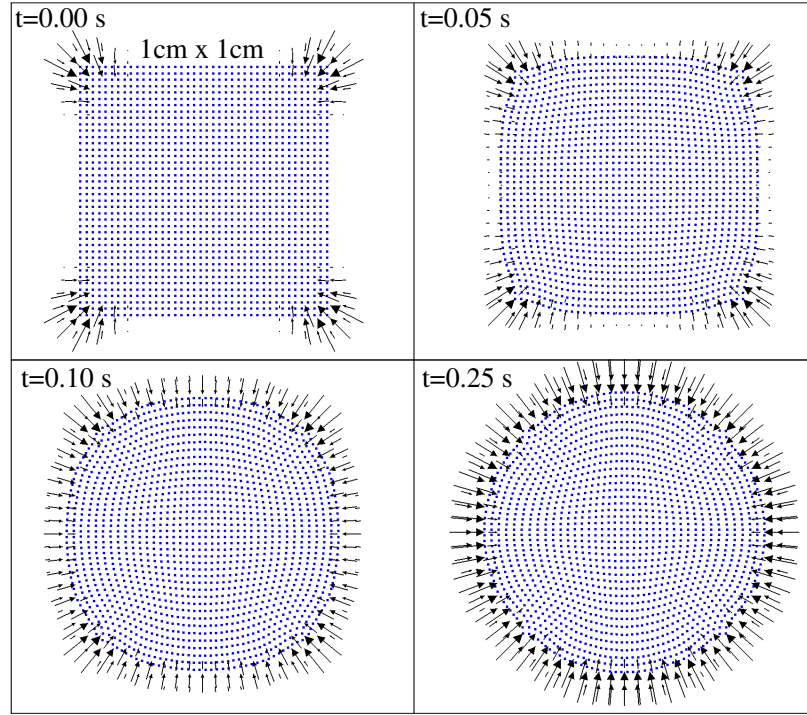


Figure 3.10: A 2D  $1 \times 1$  cm square drop evolves into a circular shape under surface tension using two iterations of moving average smoothing. The arrows represent the surface tension force.

ground grid of  $40 \times 40$  cells with a  $\text{PPC}$  of  $2\times 2$ . Similar to the previous example, there is a single layer of particles and cells in the  $z$  direction. Initially, the surface tension forces, indicated by the arrows, are concentrated at the corners where the curvatures are very large; these forces cause the drop to evolve reaching an equilibrium state at  $t = 0.25$  s. The the-

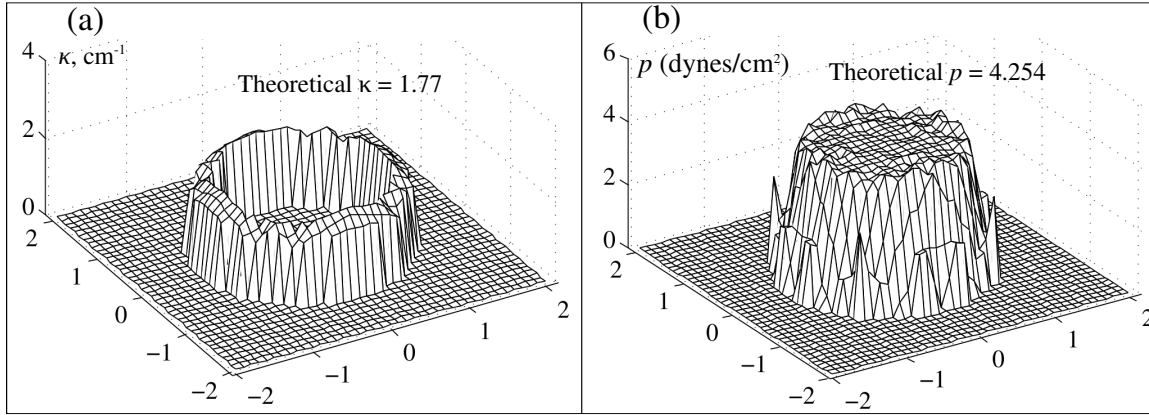


Figure 3.11: Distribution of the calculated curvature (a), and pressure (b) after a 2D non-equilibrium drop becomes stable at time = 0.25 s. The average pressure is 4.09 (or 96.1% of the theoretical value), with an rms error  $L_2 = 0.044$ .

oretical equilibrium state is circular with an area equal to that of the initial square,  $1 \text{ cm}^2$ , which gives a theoretical radius of  $0.56 \text{ cm}$ , and a curvature of  $1.77 \text{ cm}^{-1}$ ; the pressure drop across the interface is  $4.25 \text{ dynes/cm}^2$ . Fig. 3.11 shows the calculated curvature and pressure at  $t=0.25 \text{ s}$  when equilibrium is reached; the average pressure  $\langle p \rangle$  is  $4.09 \text{ dynes/cm}^2$  (or 96.1% of the theoretical value) with an rms error  $L_2=0.044$ . A similar example was studied in Brackbill et al. [1992] but the equilibrium state was not shown.

The same problem of a non-equilibrium drop evolving to its equilibrium state is simulated in 3D shown in Fig. 3.12. In this simulation, the drop is initialized as a  $1 \times 1 \times 1\text{-cm}$  cube placed over a  $2 \times 2 \times 2\text{-cm}$  background mesh. The drop is discretized into a total of 27000 particles and the background mesh has  $30 \times 30 \times 30$  cells; the PPC is  $2 \times 2 \times 2$ . The theoretical radius of the final sphere is  $0.62 \text{ cm}$  corresponding to a curvature of  $1.61 \text{ cm}^{-1}$ , and a pressure drop of  $7.74 \text{ dynes/cm}^2$  (cf. Eq. 3.30). The calculated curvature and pressure on a plane through the origin are shown in Fig. 3.13; the average pressure is  $8.05 \text{ dyn/cm}^2$  (or 104.1 % of the theoretical value) with an rms error of 0.0451. The results suggest that the GIMP-CSF algorithm is capable of handling 3D droplet simulation with good accuracy.

In summary, the proposed GIMP-CSF algorithm can produce comparable results as benchmarked by examples available in the literature or with known theoretical solutions for a drop evolving to its equilibrium state in 2D or 3D.

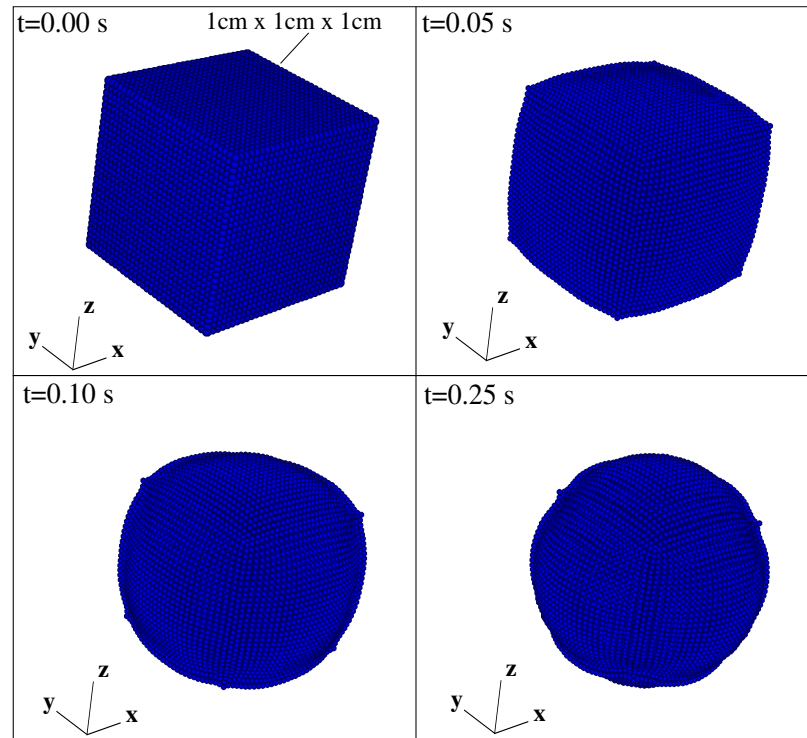


Figure 3.12: Evolution of a 3D cubic drop into a sphere under surface tension using two iterations of moving average smoothing.

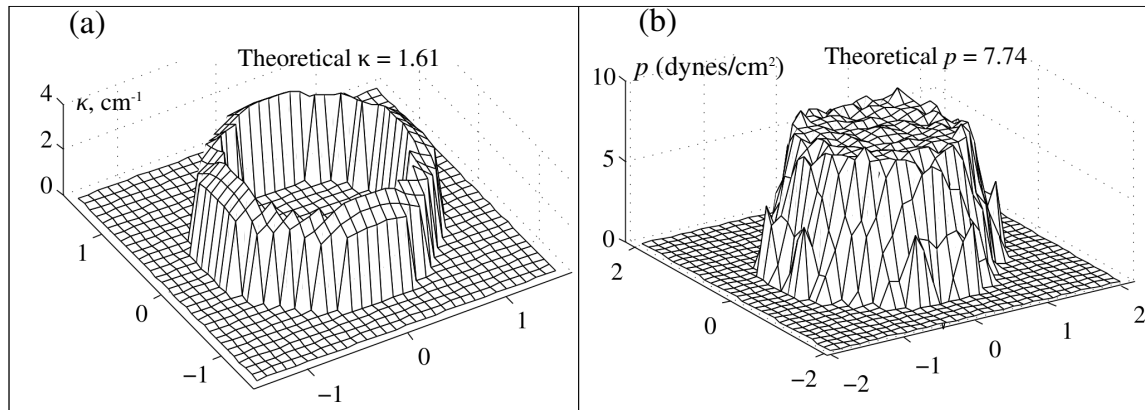


Figure 3.13: Distribution of the calculated curvature (a), and pressure (b) on a plane through the origin of a 3D drop. The average pressure is 8.05 (or 104.1% of the theoretical value) with an rms error  $L_2 = 0.0451$ .

### 3.5.3 Wall adhesion and capillary rise

The last example demonstrates the capability of the GIMP-CSF algorithm in problems involving contact angles and the gravitational force. The contact angle problem was modeled

in Brackbill et al. [1992] using the CSF method but without quantitative comparisons with theoretical results. In this example, we simulate the problem of half-plane capillary rise and compare the results with theoretical solutions.

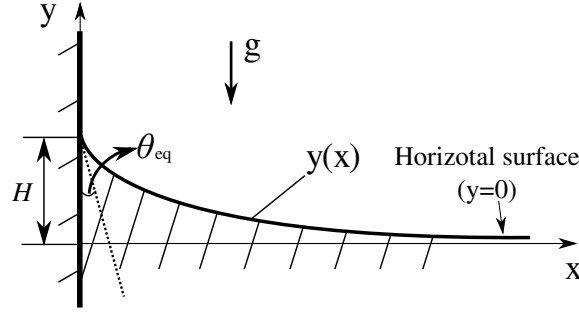


Figure 3.14: Schematic of half-plane liquid meniscus  $y(x)$  under gravitational pull. The liquid is at rest in contact with a hydrophilic wall with a given contact angle  $\theta_{eq}$ .  $H$  is the capillary rise to be simulated.

The problem is shown schematically in Fig. 3.14 where a contact angle  $\theta_{eq}$  is specified. The meniscus profile can be described by a closed-form solution derived from the work in Anderson et al. [2006]:

$$\xi = \cosh^{-1} \left( \frac{2}{\zeta} \right) - \sqrt{4 - \zeta^2} + \sqrt{2(1 - \sin \theta)} - \cosh^{-1} \left( \frac{\sqrt{2}}{\sqrt{1 - \sin \theta}} \right) + \sqrt{2 + 2 \sin \theta} \quad (3.33)$$

where the two dimensionless parameters are  $\xi = x / \sqrt{\sigma / \rho g}$ ,  $\zeta = y / \sqrt{\sigma / \rho g}$ ;  $x$  is distance measured from the wall, and  $g$  the gravitational constant ( $981 \text{ cm/s}^2$  in the negative- $y$  direction). The capillary rise  $H$  is measured from the horizontal surface at  $y = 0$  shown in Fig. 3.14.

The initial set-up of the capillary-rise simulation is shown in Fig. 3.15 where a 2D tank (0.5 cm by 0.3 cm) is partially filled with a 0.20-cm-deep fluid. A  $75 \times 45 \times 1$  background mesh with 6750 particles are used (PPC=2×2). It should be noted that the normal vector to the initial meniscus, as shown in Fig. 3.15, points to the negative  $y$ -direction at all the grid nodes lying on the meniscus except at the boundaries which join the walls. At the two side boundaries, the vectors are set in the direction defined by the contact angle  $\theta_{eq}$  as a boundary condition (cf. Section 3.4.3). Such a boundary condition results in a net surface tension force which drives the fluid upwards along the wall, while under the influence of

gravitational pull, until an equilibrium state is established.

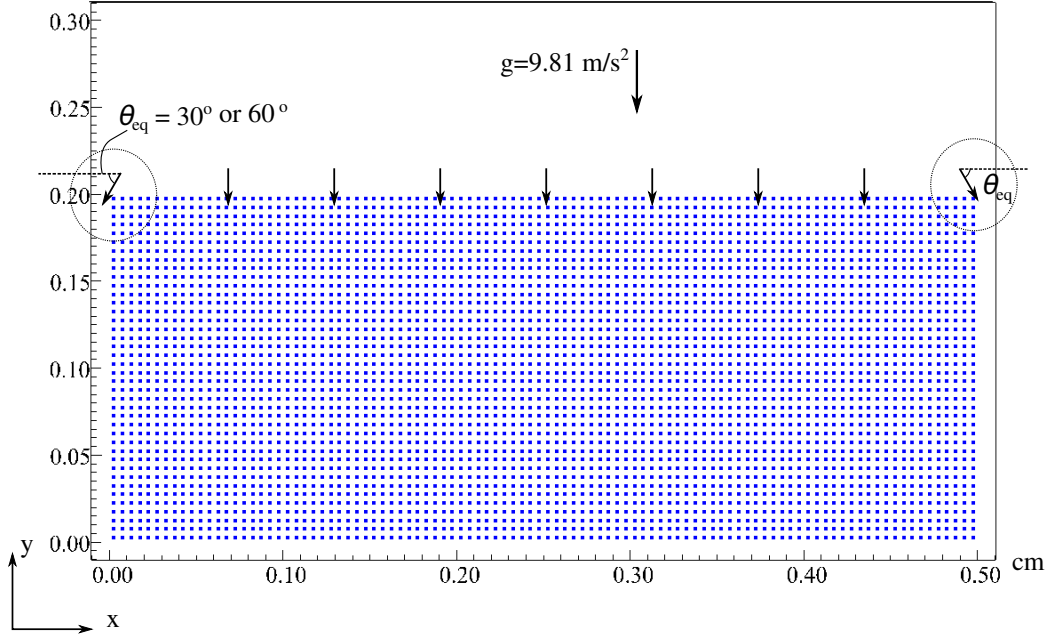


Figure 3.15: Set-up for the simulation of the capillary rise. The contact angle  $\theta_{eq}$  is prescribed at the wall boundary; the surface profile will evolve toward its equilibrium state under surface tension.

The evolution of the meniscus profile is shown in Fig. 3.16 with a contact angle gradually approaching the specified  $\theta_{eq}$  of  $30^\circ$ . The simulation is terminated when the contact angle is equal to the specified  $30^\circ$ , shown in Fig. 3.16, at  $t=0.1$  s. The profile of the meniscus is obtained using particle positions.

We compare the simulation results for two different  $\theta_{eq}$  values,  $30^\circ$  and  $60^\circ$ , with the theoretical solutions (Eq. 4.38) in Fig. 3.17. The agreement with the theoretical solutions for various  $\theta_{eq}$  is good suggesting that the proposed GIMP-CSF algorithm is capable of handling problems with wetting or capillary rise such as liquid extraction in oil recovery or capillary driven flows.

### 3.6 Discussions and Conclusion

Surface tension exists at the interface between two immiscible fluids and becomes significant at small spatial scales. In this paper, we presented a new particle-based algorithm, GIMP-CSF, to model surface tension and capillary rise in the presence of contact angles. The contribution of this paper is in the adaptation of the CSF surface tension model for the



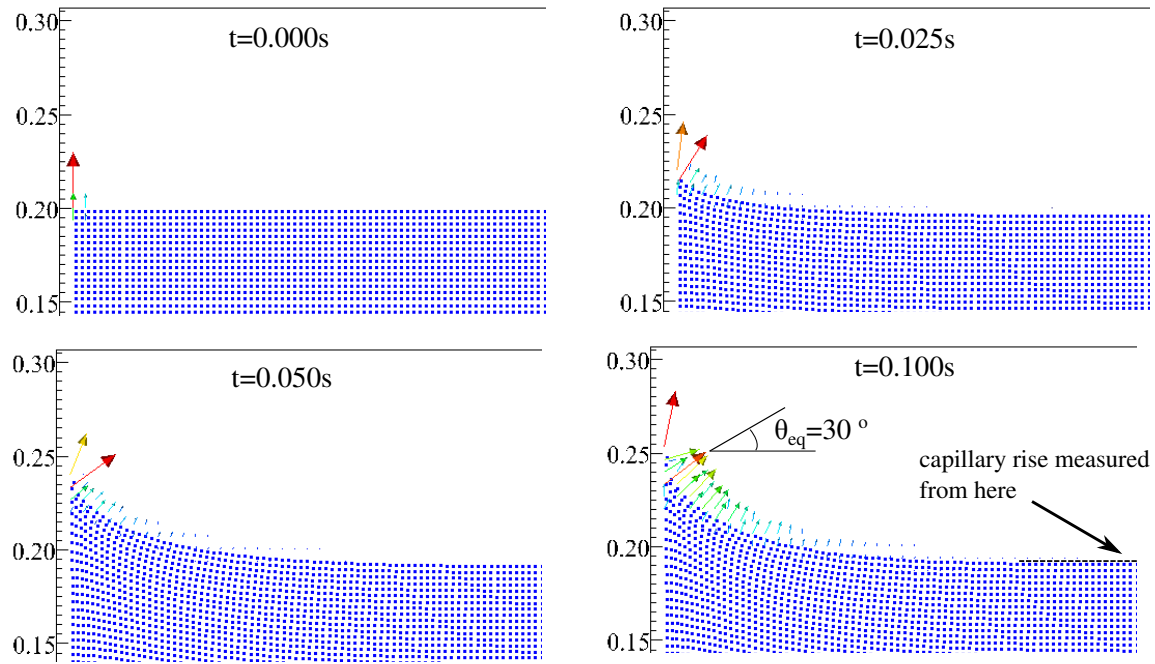


Figure 3.16: Evolution of the surface profile due to surface tension. Arrows indicate the surface tension force which are in equilibrium with gravitational pull at  $t=0.1$  s ( $\theta_{eq} = 30^\circ$ ).

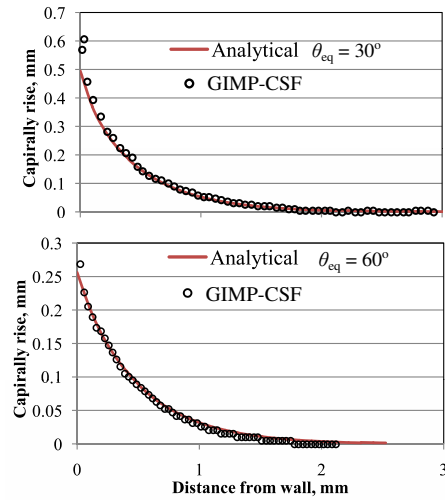


Figure 3.17: Comparison of the calculated capillary rise at  $t=0.1$  s to theoretical solutions for different contact angles ( $30^\circ$  and  $60^\circ$ ).

GIMP method. One key aspect of the algorithm is about calculating and adding the surface tension force to the equations of motion at the nodes of the background mesh within

the framework of the GIMP method. The algorithm used a moving average smoothing scheme for the color function to improve the accuracy of the curvature at the interface. The effects of smoothing, grid resolution, and number of particles per cell were also studied with practical suggestions regarding a balance between accuracy and computational efficiency.

The computational time for each example is summarized in Tab. 3.3 based upon the Uintah package installed on a Sun Ultra 40 computer with a 2.4 GHz CPU.

Table 3.3: Summary of computational time.  $\Delta h$  is mesh size,  $t_p$  is physical time, and  $t_c$  is computational time.

Example	Mesh	$\Delta h$ (cm)	# of particles	$t_p(s)$	$t_c(s)$
2D-equilibrium	40x40x1	0.1	1264	0.25	195
2D-equilibrium	40x40x1	0.1	5024	0.25	482
2D-equilibrium	80x80x1	0.05	5024	0.25	1304
2D-nonequilibrium	40x40x1	0.1	1600	0.25	571
3D-nonequilibrium	30x30x30	0.13	27000	0.25	4612
Capillary rise	75x45x1	0.0067	6750	0.1	4556

As demonstrated through the benchmarked examples, we have shown that the proposed GIMP-CSF algorithm, using a nearly-incompressible constitutive law for fluids, can be as effective and accurate as other methods in literature. This paves the way for applying GIMP toward more difficult problems such as fluid-structure interactions at small scales where surface tension plays an important role.

### 3.7 References

- M. L. Anderson, A. P. Bassom, and N. Fowkes. Exact solutions of the Laplace-Young equation. *Proceedings of the Royal Society A*, 462:3645–3656, 2006.
- S. Bardenhagen and E. Kober. The generalized interpolation material point method. *CMES: Computer Modeling in Engineering and Sciences*, 5:477–495, 2004.
- S. Bardenhagen, J. Guilkey, K. Roessig, J. Brackbill, W. Witzel, and J. Foster. An improved contact algorithm for the material point method and application to stress propagation in granular material. *CMES: Computer Modeling in Engineering and Sciences*, 2:509–522, 2001.
- F. Blanchette and T. P. Bigioni. Dynamics of drop coalescence at fluid interfaces. *J. Fluid Mech.*, 620:333–352, 2009.
- C.D. Boor. *A Practical Guide to Splines*. Springer-Verlag, New York, 1967.
- J. U. Brackbill, D.B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100:335–354, 1992.
- J.U. Brackbill and H.M. Ruppel. FLIP: a method for adaptively zoned, particle-in-cell calculation of fluid flows in two dimensions. *Computer Physics Communications*, 95:314–343, 1986.
- M. Bussmann, J. Mostaghimi, and S. Chandra. On a three-dimensional volume tracking model of droplet impact. *Physics of Fluids*, 11:1406–1418, 1999.
- Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *Journal of Computational Physics*, 124:449–464, 1996.
- L. Cueto-Felgueroso, G. Mosqueira I. Colominas, F. Navarrina, and M. Casteleiro. On the Galerkin formulation of the smoothed particle hydrodynamics method. *International Journal for Numerical Methods in Engineering*, 60:1475–1512, 2004.
- Y. Gan, Z. Chen, and S. Montgomery-Smith. Improved material point method for simulating the zona failure response in piezo-assisted intracytoplasmic sperm injection. *CMES: Computer Modeling in Engineering and Sciences*, 73:45–75, 2011.

- Denis Gueyffier, Jie Li, Ali Nadim, Ruben Scardovell, and Stephane Zaleski. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics*, 152:423–456, 1999.
- J.E. Guilkey, T.B. Harman, and B. Banerjee. An Eulerian-Lagrangian approach for simulating explosions of energetic devices. *Computers and Structures*, 85:660–674, 2007.
- X.Y. Hu and N.A. Adams. A constant-density approach for incompressible multi-phase SPH. *Journal of Computational Physics*, 228:2082–2091, 2009.
- J. H. Lee and J. E. Guilkey. Grain-scale modeling of saturated soils using the generalized interpolation material point method. In *Proceedings of the 16th International Conference of the International Society for Terrain-Vehicle Systems*, 2008.
- F. Li, J. Pan, and C. Sinka. Modeling brittle impact failure of disc particles using material point method. *International Journal of Impact Engineering*, 38:653–660, 2011.
- S. Li and W. K. Liu. Meshfree and particle methods and their applications. *Applied Mechanics Review*, 55:1–34, 2002.
- X.-D. Liu, R.P. Fedkiw, and M. Kang. A boundary condition capturing method for multi-phase incompressible flow. *Journal of Scientific Computing*, 160:151–178, 2000.
- J. J. Monaghan, R. A. F. Cas, A. M. Kos, and M. Hallworth. Gravity currents descending a ramp in a stratified tank. *Journal of Fluid Mechanics*, 379:39–70, 1999.
- J.P. Morris. Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 33:333–353, 2000.
- J.P. Morris, P.J. Fox, and Y. Zhu. Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics*, 136:214–226, 1997.
- J. A. Nairn. Material point method calculations with explicit cracks. *CMES: Computer Modeling in Engineering and Sciences*, 4:649–663, 2003.
- H. O. Nordmark. Rezoning for higher order vortex methods. *Journal of Computational Physics*, 97:366–397, 1991.
- S. Nugent and H. A. Posch. Liquid drops and surface tension with smoothed particle applied mechanics. *Physical Review E*, 62:4968–4975, 2000.

- S. Parker, J. Guilkey, and T. Harman. A component-based parallel infrastructure for the simulation of fluid-structure interaction. *Engineering with Computers*, 22:277–292, 2006.
- M. Steffen, P.C. Wallstedt, J.E. Guilkey, R.M. Kirby, and M. Berzins. Examination and analysis of implementation choices within the material point method (MPM). *CMES: Computer Modeling in Engineering and Sciences*, 31:107–127, 2008.
- D. Sulsky, Z. Chen, and H.L. Schreyer. A particle method for history dependent materials. *Computer Methods in Applied Mechanics and Engineering*, 118:179–196, 1994.
- M. Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, 187:110–136, 2003.
- M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.
- P.C. Wallstedt and J.E. Guilkey. An evaluation of explicit time integration schemes for use with the generalized interpolation material point method. *Journal of Computational Physics*, 227:9628–9642, 2008.
- M.W. Williams, D.B. Kothe, and E.G. Puckett. Convergence and accuracy of continuum surface tension models. *Fluid Dynamics at Interface*, pages 294–305, 1999.
- A. R. York, D. L. Sulsky, and H. L. Schreyer. The material point method for simulation of thin membranes. *International Journal for Numerical Methods in Engineering*, 44:1429–1456, 1999.
- M. Y. Zhang. Simulation of surface tension in 2D and 3D with smoothed particle hydrodynamics method. *Journal of Computational Physics*, 229:7238–7259, 2010.

## Chapter 4

### Using the Generalized Interpolation Material Point Method for Fluid-Solid Interactions Induced by Surface Tension<sup>1</sup>

#### 4.1 Introduction

Fluid-structure interactions (FSI) can take place at small scales in unsaturated soil or porous rock, as well as in micro/nano manipulation processes. In those situations, the effects of surface tension and contact angle are important in determining the interactions between fluids and solids. The studies of fluid-solid interactions dominated by surface tension are of practical interest for many industrial processes, including oil recovery and chemical assay by lab-on-a-chip devices [Kapur and Gaskell, 2007].

Conducting analytical analysis on FSI problems involving surface tension can be very difficult due to the complexity of the topology of fluid interface and the contact conditions at fluid-solid interface. Thus, numerical approaches are often sought to solve these problems.

For solving common FSI problems without surface tension, meshed methods have been developed. These methods include the immersed boundary method [Xu and Wang, 2008, Gil et al., 2013], Lattice Boltzmann method [Shin et al., 2013], arbitrary Lagrangian-Eulerian (ALE) method, and level set method [Basting and Weismann, 2013]. These methods often encounter mesh entangling issues in solving large deformation problems. Besides, these methods could have issues in handling computational frameworks for both solids and fluids due to their strategy in adopting different solvers for the solid and fluid phases.

On the other hand, the mesh-free or particle methods, have also been developed to solve FSI problems. These methods, which include smoothed-particle hydrodynamics (SPH) method and material point method (MPM), are free of mesh entangling. Example of SPH in simulating FSI problems can be found in Lahooti et al. [2011]. Modeling of surface tension and contact angle using SPH can be found in Das and Das [2010]. Compared to SPH, MPM is a relatively new method developed by Sulsky et al. [1994, 1995] by extending the hydrodynamics fluid implicit particle (FLIP) method [Brackbill and Ruppel, 1986] to solid mechanics. Although MPM is mainly used for solid mechanics problems, it has also been applied to FSI problems such as membrane-gas interactions by York et al.

---

<sup>1</sup>A version of this chapter is prepared for submission to Journal of Computational Physics as L. Chen, J.H. Lee, and C.-f. Chen. Using the Generalized Interpolation Material Point Method for Fluid-Solid Interactions Induced by Surface Tension.

[1999], and water-soil interactions by Lee and Guilkey [2008].

Recently, surface tension has been implemented using the generalized interpolation material point (GIMP) method by Chen et al. [2012] where surface tension force is incorporated into the solver. For the first time, GIMP is able to solve surface tension problems. This revised GIMP method is thus named as GIMP-CSF. The capability of GIMP to simulate problems of computational fluid dynamics(CFD) at low Reynold's number, on the other hand, has been systematically evaluated by Chen et al. [2013], referred to as GIMP-CFD. Building upon the capabilities of GIMP-CSF and GIMP-CFD, it is of interest to study the feasibility of GIMP in solving FSI problems induced by surface tension at small scales where the Reynolds number of fluids is small. One unique feature of MPM is that a single computational framework is used for both the fluid and solid phases.

In this paper, we present a new method for solving the FSI problems dominated by surface tension. Our method integrates GIMP-CFD and GIMP-CSF for general fluid dynamics as well as surface tension forces.

The paper is organized as follows. We first review the concept of surface tension and contact angle in Section 4.2 which is crucial to model properly FSI problems induced by surface tension. In Section 4.3, we review the basic equations of MPM where surface tension forces are incorporated into the governing equations. In Section 4.4, the surface tension model is proposed to include the effects of surface tension, as well as contact angle on both fluid and solid. Section 4.5 describes the contact algorithm for modeling the partial-slip boundary condition at the triple point where the solid, liquid, and gas phases meet. Numerical examples are presented in Section 4.6 in which three examples are used to verify the proposed algorithm. Discussion and conclusions are given in the end.

## 4.2 Review of surface tension and contact angle

Surface tension plays an important role in FSI problems at small scales. An appropriate description of the fluid-solid contact situation is key in understanding FSI problems such as water retention in unsaturated soils, and material deformation in micro/nano manipulations.

Figure 4.1 shows a droplet resting on a solid substrate to illustrate the concept of surface tension and contact angle. The state of equilibrium of this system can be described by force

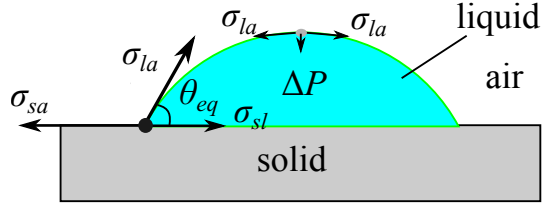


Figure 4.1: Schematic of surface tension and contact angle for a liquid-air-solid system. Fluid is subject to capillary pressure  $\Delta P$ , while solid is subject to capillary pressure and the attraction force equal to  $\sigma_{la} \sin \theta_{eq}$ .

equilibrium in the horizontal direction:

$$\sigma_{la} \cos \theta_{eq} = \sigma_{sa} - \sigma_{sl} \quad (4.1)$$

where  $\sigma_{la}$ ,  $\sigma_{sa}$ , and  $\sigma_{sl}$  represent the liquid-air, solid-air, and solid-liquid interfacial tensions, respectively, and  $\theta_{eq}$  is the contact angle. The pressure difference,  $\Delta P$ , across the fluid interface is dependent on the interface curvature  $\kappa$  described by the Young-Laplace equation [Young, 1805]:

$$\Delta P = \sigma_{la} \kappa \quad (4.2)$$

The capillary force imposed on the solid is attributed to the Laplace pressure force exerted from within the liquid, and the surface tension force exerted from the contact line of three-phase interaction which can be expressed by:

$$F_{cap} = \Delta P \cdot A + \sigma_{la} l_c \sin \theta_{eq} \quad (4.3)$$

where  $A$  is the area of contact between the liquid and solid phases. Eqs. 4.1-4.3 are the key ones for the accurate simulation of FSI problems induced by surface tension. Both capillary pressures and forces must be modeled correctly by the computer code in order to obtain the true interaction between the solid and the fluid, the goal of this paper.

### 4.3 Basic equations of MPM

To develop the algorithm of GIMP-FSI, we first review basic equations used in MPM procedures. According to conservation of mass and momentum, the following governing



equations are solved in MPM for both fluid and solid materials:

$$\frac{d\rho}{dt} + \rho \nabla \cdot \mathbf{v} = 0 \quad (4.4)$$

and

$$\rho \mathbf{a} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} \quad (4.5)$$

where  $\rho$  is the density,  $\mathbf{v}$  is the velocity,  $\boldsymbol{\sigma}$  is the stress tensor,  $\mathbf{a}$  is the acceleration,  $\mathbf{b}$  is the specific body force. Using the same governing equations for both solid and fluid materials in MPM is a key difference from the meshed-based solvers for FSI problems.

For illustration of MPM procedures, Fig. 4.2 shows a two-dimensional, one-solid and

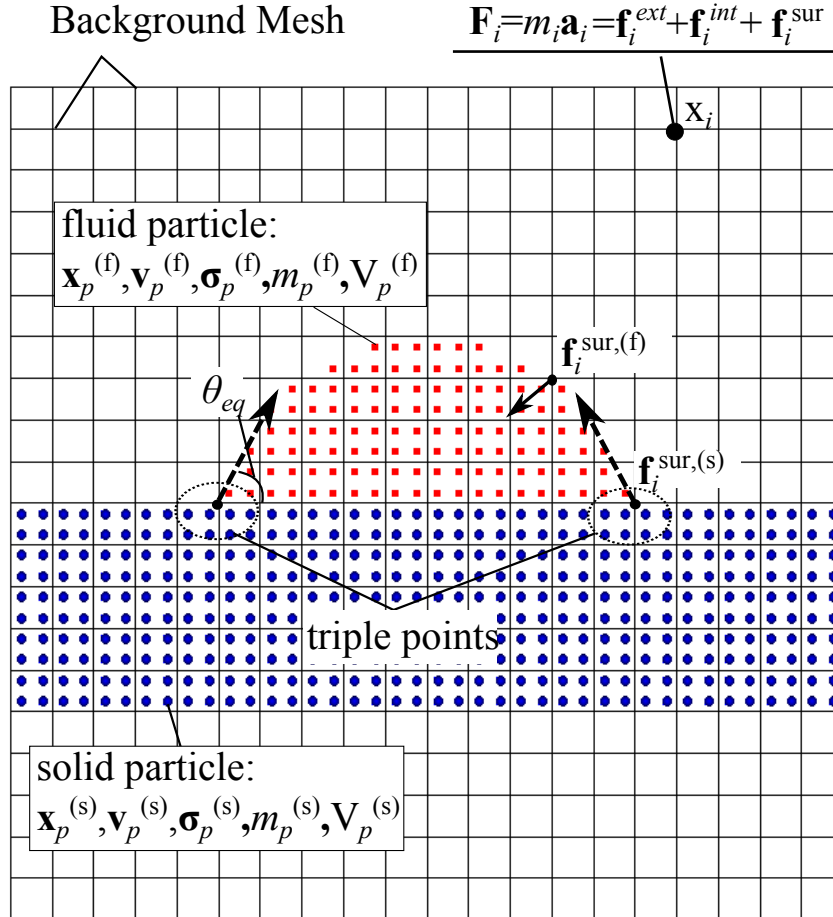


Figure 4.2: Example of GIMP-FSI.  $\mathbf{x}_p$ -particle position;  $\mathbf{v}_p$ -particle velocity;  $\boldsymbol{\sigma}_p$ -particle stress;  $V_p$ -particle volume;  $\mathbf{f}_i^{sur,(f)}$ -surface tension force on grid nodes acting around fluid interface;  $\mathbf{f}_i^{sur,(s)}$ -surface tension force acting on the solid near the triple points.

one-fluid system with the fluid resting on the solid plate. We use superscript “f” to denote fluid variables, and superscript “s” for solid variables. To solve Eqs. 4.4-4.5, the fluid and solid objects are discretized by a number of particles with fixed mass ( $m_p$ ). The particles carry all the state variables of each material, including particle position  $\mathbf{x}_p$ , velocity  $\mathbf{v}_p$ , stress  $\boldsymbol{\sigma}_p$ , deformation gradient  $\mathbf{F}_p$  and volume  $V_p$ . Since particle mass  $m_p$  is fixed, Eq. 4.4 representing the conservation of mass is automatically satisfied.

For solving Eq. 4.5, a background mesh, usually a regular Cartesian grid for computational convenience, is used in MPM as a scratch pad, a key difference from other particle methods such as SPH. On each grid node  $i$ , the following equation, derived from Eq. 4.5, is solved for each material:

$$m_i \mathbf{a}_i = \mathbf{f}_i^{ext} - \mathbf{f}_i^{int} + \mathbf{f}_i^{sur} \quad (4.6)$$

in which  $m_i$  denotes the grid mass,  $\mathbf{a}_i$  the grid acceleration;  $\mathbf{f}_i^{ext}$  represents the grid external force,  $\mathbf{f}_i^{int}$  is the grid internal force, and  $\mathbf{f}_i^{sur}$  is the surface tension forces that are required for solving FSI problems at small scales.

Since all the state variables are stored in particles, solving Eq. 4.6 requires mapping between particles and grid nodes via an appropriate interpolation function  $S_{ip}$  (e.g., GIMP interpolation function). The grid variables needed in Eq. 4.6 can be obtained by the following equations (for both fluid and solid):

$$m_i = \sum_p S_{ip}(\mathbf{x}_p) m_p \quad (4.7)$$

$$\mathbf{f}_i^{int} = M_p \sum_p \mathbf{G}_i(\mathbf{x}_p) \cdot \boldsymbol{\sigma}_p^{sp} \quad (4.8)$$

$$\mathbf{f}_i^{ext} = \sum_p S_{ip}(\mathbf{x}_p) \mathbf{b}_p + \sum_p S_{ip}(\mathbf{x}_p) \boldsymbol{\tau}_p \quad (4.9)$$

where  $\sum_p$  denotes summation over particles,  $\mathbf{G}_i(\mathbf{x}_p) = \nabla S_{ip}$  is the gradient of the interpolation function,  $\boldsymbol{\sigma}_p^{sp}$  is the specific stress tensor,  $\mathbf{b}_p$  is particle body force, and  $\boldsymbol{\tau}_p$  is particle traction force.

For FSI problems, surface tension forces are also considered in solving equations of motion. As shown in Fig. 4.2, the fluid is subject to a surface tension force denoted by  $\mathbf{f}_i^{sur,(f)}$  at the fluid interface, and the solid is subject to  $\mathbf{f}_s^{sur,(f)}$  at the triple points. Calculations of the two forces are given in Section 4.4.1 and Section 4.4.3, respectively.

Once Eq. 4.6 is solved, a new grid velocity,  $\mathbf{v}_i^L$ , can be calculated from the grid acceler-

ation for a time increment  $\Delta t$ :

$$\mathbf{v}_i^L = \mathbf{v}_i + \mathbf{a}_i \Delta t \quad (4.10)$$

in which  $\mathbf{v}_i$  is the old grid velocity given by

$$\mathbf{v}_i = \frac{\sum_p S_{ip}(\mathbf{x}_p) \mathbf{v}_p M_p}{m_i} \quad (4.11)$$

With  $\mathbf{a}_i$  and  $\mathbf{v}_i^L$  calculated on a grid node, particle variables can be updated for the next computational cycle by mapping the new grid variables back via the interpolation function  $S_{ip}$ :

$$\Delta \mathbf{v}_p = \sum_i S_{ip} \mathbf{a}_i \Delta t \quad (4.12)$$

and the incremental particle position is given by

$$\Delta \mathbf{x}_p = \sum_i S_{ip} \mathbf{v}_i^L \Delta t \quad (4.13)$$

With appropriate constitutive law for each material, the particle stress is updated by:

$$\Delta \mathbf{F}_p = \mathbf{I} + \Delta t \sum_i \mathbf{G}_{ip} \mathbf{v}_i^L \quad (4.14)$$

and

$$\Delta \boldsymbol{\sigma}_p = f(\Delta \mathbf{F}_p) \quad (4.15)$$

where  $\Delta \mathbf{F}_p$  is the increment of the deformation gradient tensor,  $\mathbf{I}$  the identity tensor, and  $f(\Delta \mathbf{F}_p)$  describes the constitutive law of each material. In Eq. 4.14, the stress is updated using the new grid velocity  $\mathbf{v}_i^L$ , which results in better performance than using the old velocity  $\mathbf{v}_i$  [Wallstedt and Guilkey, 2008].

Note that, unwanted vibration in a dynamic system can take a long time for the system to stabilize. In order to speed up computation, artificial damping can be added to the grid acceleration Chen et al. [2013]:

$$\mathbf{a}_i^{damp} = \mathbf{a}_i - \xi \mathbf{v}_i^L \quad (4.16)$$

where  $\xi$  is the damping coefficient, the value of which is chosen by the user suitable for the problem at hand.

#### 4.4 Surface tension model in GIMP-FSI

##### 4.4.1 The CSF model for fluid

Solving the governing equations in Eq. 4.6 requires an appropriate surface tension model for calculating the surface tension forces. To model surface tension forces for fluid, we adopt the continuum surface force method (CSF) proposed by Brackbill et al. [1992], which has been implemented in GIMP-CSF by Chen et al. [2012]. In GIMP-CSF, the following equation was proposed to calculate the surface tension force,  $\mathbf{f}_i^{sur,(f)}$  (see Fig. 4.2):

$$\mathbf{f}_i^{sur,(f)} = V_c \frac{\sigma \kappa}{[m] < m >^{(f)}} (\nabla m_i^f), \quad (4.17)$$

in which  $V_c$  is cell volume,  $\sigma$  is surface tension coefficient of liquid-air interface (the superscript “*la*” is omitted from now on),  $\kappa$  is curvature of fluid interface,  $[m]$  and  $< m >$  are two constants based on fluid density, and  $m_i$  is fluid mass determined by Eq. 4.7. The curvature  $\kappa$  used in Eq. 4.17 can be calculate by:

$$\kappa = \nabla \cdot \tilde{\mathbf{n}}^f = \nabla \cdot \frac{\nabla \tilde{m}_i^f}{|\nabla \tilde{m}_i^f|} \quad (4.18)$$

in which  $\tilde{\mathbf{n}}^f$  is a smoothed fluid normal calculated by the smoothed fluid mass  $\tilde{m}_i$ . Eqs. 4.17-4.18 proved to be effective in simulating a single fluid with surface tension as reported by Chen et al. [2012].

When a solid-fluid interface forms, difficulties arise in using Eq. 4.18 to calculate the curvature near the triple points. The main reason is that smoothing of fluid mass may not be applicable at triple points where the fluid interface changes sharply. This issue can be demonstrated in Fig. 4.3a, which shows that the use of smoothed fluid mass  $\tilde{m}_i^f$  will result in a greater error when calculating the curvature at the triple points. In addition, when the smoothed fluid mass is used to calculate surface normals, surface normals at the interface of two immiscible phases – such as the fluid-solid interface – will also arise thus inducing surface tension force. However, in physics, surface tension should not exist at the fluid-solid interface except at the triple points.

To overcome the difficulty of GIMP-CSF in dealing with the triple points, an alternative approach for estimating the curvature of fluid interface is proposed here shown in Fig. 4.3b. Basically, rather than using smoothed fluid mass, the smooth normal  $\tilde{\mathbf{n}}^f$  for cur-

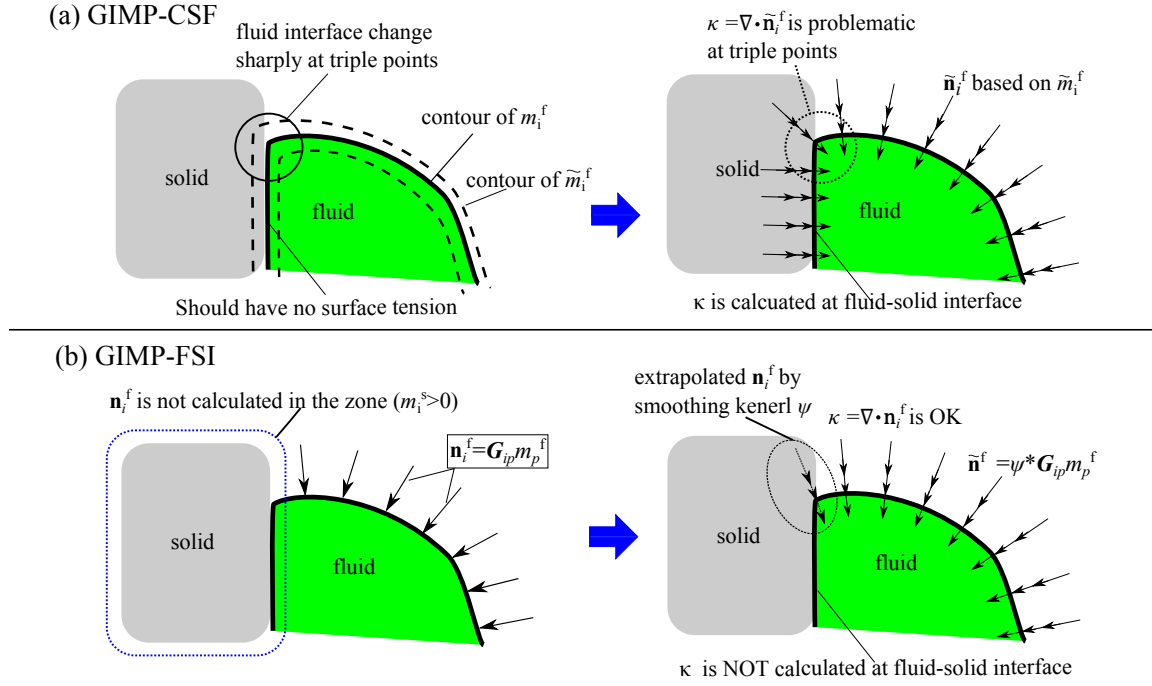


Figure 4.3: a) Curvature  $\kappa$  calculated by divergence of  $\tilde{\mathbf{n}}^f$  from the smoothed fluid mass is problematic at the triple points; (b) curvature  $\kappa$  computed by divergence of the smoothed  $\tilde{\mathbf{n}}^f = \mathbf{G}_{ip} m_p^f$  with a smoothing kernel  $\psi$ .  $\mathbf{G}_{ip} m_p^f$  is not calculated where solid mass ( $m_i^s$ ) is non-zero to avoid the sudden change of interface at the triple points.

vature calculation is computed via a smoothing kernel  $\psi$  :

$$\tilde{\mathbf{n}}_i^f = \psi * \mathbf{n}_i^f \quad (4.19)$$

$$\mathbf{n}_i^f = \frac{\sum_p G_{ip} m_p^f}{|\sum_p G_{ip} m_p^f|} \quad (m_i^s = 0) \quad (4.20)$$

in which the fluid normal is only calculated on the grid nodes without the solid mass. Various smoothing kernels  $\psi$  can be used in Eq. 4.19, and examples can be found in Meier et al. [2002]. According to Kalland [2008], the following smoothing kernel gives better results for curvature estimation in 2D:

$$\psi = [1, 2, 1, 2, 4, 2, 1, 2, 1] \quad (4.21)$$

and

$$\psi = [1, 2, 1, 2, 4, 2, 1, 2, 1], [2, 4, 2, 4, 8, 4, 2, 4, 2], [1, 2, 1, 2, 4, 2, 1, 2, 1] \quad (4.22)$$

for 3D smoothing. Note that in Eq. 4.20, the fluid normal is tapered by a zone influenced by solid in order to avoid the sudden change of the fluid interface at the triple points as well as the fluid-solid interface.

The smoothed fluid normal  $\tilde{\mathbf{n}}^f$  covers a wider region than the original  $\mathbf{n}^f$ , but may not cover the triple points where surface tension force should also be calculated. To ensure that the triple points are also covered by  $\tilde{\mathbf{n}}^f$ , one can use the smoothing kernel more than once, so that:

$$\tilde{\mathbf{n}}_i^f = \psi^{(n)} * \mathbf{n}_i^f \quad (4.23)$$

in which  $\psi^{(n)}$  denotes that the smoothing kernel  $\psi$  is applied  $n$  times. For curvature calculation, using  $\psi^{(2)}$  ( $n=2$ ) is usually sufficient, and more smoothing may result in an interface to be too far from the true interface.

In summary, GIMP-FSI adopts similar equations as GIMP-CSF to solve the surface tension forces of fluid except for the difference in curvature calculation. The new technique of the curvature calculation is devised to avoid the sharp interface change at the triple points and skip calculation of surface tension along the fluid-solid interface.

#### 4.4.2 Treatment of contact angle

According to the Young-Laplace equation (Eq. 4.1), the contact angle is a result from the energy balance of three phases at the triple points. In GIMP-FSI, the effect of contact angle can be achieved by following the CSF method, which replaces the fluid normal  $\tilde{\mathbf{n}}^f$  at the solid interface with  $\tilde{\mathbf{n}}_{eq}$ :

$$\tilde{\mathbf{n}}_{eq} = \mathbf{n}^s \cos \theta_{eq} + \mathbf{t}^s \sin \theta_{eq} \quad (4.24)$$

where the solid normal  $\mathbf{n}^s$  and solid tangent  $\mathbf{t}^s$  can be calculated by:

$$\mathbf{n}^s = \frac{\sum_p G_{ip} m_p^s}{|\sum_p G_{ip} m_p^s|} \quad (4.25)$$

$$\mathbf{t}^s = \frac{\tilde{\mathbf{n}}^f - \tilde{\mathbf{n}}^f \cdot \mathbf{n}^s}{|\tilde{\mathbf{n}}^f - \tilde{\mathbf{n}}^f \cdot \mathbf{n}^s|} \quad (4.26)$$

Similar treatment was also implemented in GIMP-CSF, in which the boundary normal and tangent are used. Treatment of contact angle using Eq. 4.24 will force the fluid at the triple points to move until the specified contact angle is reached, which will be demonstrated in the verification examples

#### 4.4.3 Surface tension acting on solid

According to Eq. 4.3, the total capillary force exerting on solid consists of two components. The first is the capillary pressure transferred from the interacting fluid via an appropriate contact algorithm which will be described in Sec. 4.5. The second component is the tangential surface tension force at the triple points. The force, denoted here by  $\mathbf{f}^{sur,(s)}$ , can be expressed by:

$$\mathbf{f}_i^{sur,(s)} = (\sigma l_c) \mathbf{t}^f \quad (4.27)$$

in which  $l_c$  is the length of the contact line, and  $\mathbf{t}^f$  is the tangent of fluid interface at triple points. Note that for a 2D problem, the contact line is in the third direction and  $l_c$  becomes the thickness of the domain. When the fluid-solid system is at equilibrium, the angle formed between  $\mathbf{t}^f$  and the solid interface is then the so-called contact angle  $\theta_{eq}$ .

Although Eq. 4.27 is relatively simple, numerical calculation of  $\mathbf{f}^{sur,(s)}$  faces two challenging tasks. First, the positions of triple points must be accurately identified. Second, the length of contact line  $l_c$  must be calculated. We found that the accuracy of the results are sensitive to both tasks, which are not trivial in MPM.

For a 2D problem, the length of the contact line is the depth of the third direction. Therefore, it only needs to locate the triple points. We use the following criterion to identify a candidate of triple points: a triple point can't have all of its neighboring nodes lying in the fluid domain. First, we exclude nodes that have zero fluid mass and zero solid normal  $\mathbf{n}^s$  given in Eq. 4.25. Next, we examine the following condition to determine if the node is a triple point or not:

$$\text{if } m_{i,j-1}m_{i,j+1} = 0, \text{ then node } (i,j) \text{ is a triple point.} \quad (4.28)$$

in which we assume the  $x$  component of the solid normal is larger than the  $y$  component ( $|\mathbf{n}_{i,x}^s| \geq |\mathbf{n}_{i,y}^s|$ ). If  $|\mathbf{n}_{i,x}^s| < |\mathbf{n}_{i,y}^s|$ , the criterion in Eq. 4.28 becomes:

$$\text{if } m_{i-1,j}m_{i+1,j} = 0, \text{ then node } (i,j) \text{ is a triple point.} \quad (4.29)$$

Figure 4.4 shows a schematic of the search for triple points. With the steps discussed above, the search in 2D usually leads to the identification of only one node near the triple point, at which the force  $\mathbf{f}^{sur,(s)}$  in Eq. 4.27 is applied.

It is also possible that more than one node will be found around a triple point. To avoid

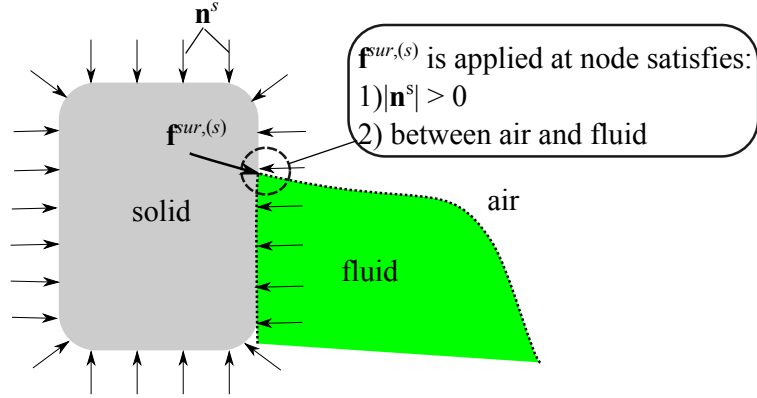


Figure 4.4: Schematic of triple point identification to apply surface tension force  $\mathbf{f}^{sur,s}$  on solid by a simple criterion.  $\mathbf{n}^s$ - solid normal;

overestimation of the surface tension effect on the solid, a  $3 \times 3$  stencil is used to calculate the average surface tension forces around each selected node:

$$\mathbf{f}_{ave}^{sur,(s)} = \frac{\sum^{3 \times 3} \mathbf{f}_{i,j}^{sur,(s)}}{N} \quad (4.30)$$

in which  $N$  is the number of nodes searched at the triple points. If there are no other triple points around the current one, then  $N = 1$ , i.e., no averaging is needed. According to our experience, the above procedure is accurate enough for 2D problems simulated in this paper. Although the same principle can be used for 3D, the inclusion of  $\mathbf{f}^{sur,(s)}$  in 3D is more difficult than 2D, and is left as future work.

#### 4.5 Contact algorithm with partial slip

One of MPM's advantages over conventional meshed methods (e.g., the finite element method) lies in the fact that the contact of multiple bodies of a single material does not need special treatment. More complicated contact situations involving friction, and multiple materials have been implemented in MPM, e.g. Bardenhagen et al. [2001]. The no-slip (single-velocity) contact is one of the simplest contact conditions for multiple materials achieved by assigning the center of mass velocity  $\mathbf{v}^{cm}$  to each material:

$$\tilde{\mathbf{v}}_i^s = \tilde{\mathbf{v}}_i^f = \mathbf{v}_i^{cm} \quad (4.31)$$

$$\mathbf{v}_i^{cm} = (m_i^f \mathbf{v}_i^f + \mathbf{v}_i^s m_i^s) / (m_i^f + m_i^s) \quad (4.32)$$



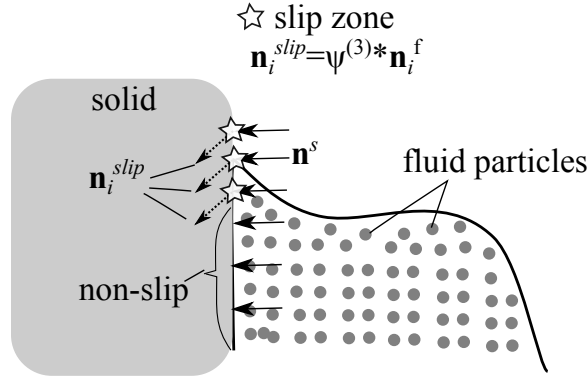


Figure 4.5: Slip of fluid near the triple points. The slip zone involves the nodes where  $\mathbf{n}^s$  and  $\tilde{\mathbf{n}}_i^{slip}$  overlap.

Generally, no-slip contact using Eq. 4.31 is applicable for most FSI problems. However, when surface tension and contact angle are involved, no-slip contact breaks down near the triple points according to Yamamoto et al. [2013]. Illustration of slip at the triple points is shown in Fig. 4.5. Because of surface tension at the triple points, fluid particles tend to slip against the wall until the static contact angle is reached. The slip near the triple points was also shown by Qian et al. [2003] using molecular dynamics simulations. It was also shown in the front-tracking simulation that a desired capillary rise can only be achieved by allowing slip near the triple points.

Taking the previous studies by Yamamoto et al. [2013] into account, a partial slip boundary condition is also implemented in GIMP-FSI by assigning a slip velocity,  $\mathbf{v}_i^{slip}$ , on the nodes near the triple points. The first step is to define a slip zone by the overlapping area of the solid normal  $\mathbf{n}^f$ , and a smoothed fluid surface normal  $\mathbf{n}_i^{slip}$ :

$$\mathbf{n}_i^{slip} = \psi^{(3)} * \mathbf{n}_i^f \quad (4.33)$$

As shown in Fig. 4.5, for the nodes in the slip zone, their velocities are replaced by a slip velocity,  $\mathbf{v}_i^{slip}$ :

$$\mathbf{v}_i^{slip} = \mathbf{v}_i - (\mathbf{v}_i^{cm} \cdot \mathbf{n}_i^s) \mathbf{n}_i^s \quad (4.34)$$

With Eq. 4.34, the fluid and solid are allowed to slip against each other at the triple points; separation, however, is not allowed. According to Lemiale et al. [2010], selecting the appropriate contact-surface normal vectors is crucial for the contact algorithm to work properly. In GIMP-FSI, we use the solid normal  $\mathbf{n}^s$  rather than the fluid normal  $\mathbf{n}^f$  as the contact

surface normal. In addition, we reject any  $\mathbf{n}^s$  when  $|\sum \mathbf{G}_{ip} m_p^s|$  is smaller than a threshold of  $\frac{\rho^s V_c}{5dx}$  ( $V_c$  is cell volume, and  $dx$  is cell size). Consequently, an effective partial-slip contact algorithm can be achieved which can prevent early enforcement of the contact as reported in Lemiale et al. [2010].

#### 4.6 Numerical examples

In this section, three different FSI problems are used to verify the proposed GIMP-FSI algorithm. The first is the half-plane capillary rise problem which was studied in a previous work by Chen et al. [2012]. With the GIMP-FSI algorithm we see an improvement which will be compared to our previous work in this section. The second problem is the modeling of a liquid bridge between two parallel plates and two cylinders of equal radii, respectively. The problem is used to show the accuracy of GIMP-FSI in estimating capillary attraction forces caused by surface tension. The third problem is to simulate the abnormal bending of a micro-beam caused by a droplet. The problem is used to demonstrate the capability of GIMP-FSI in elastocapillary problems at small scales.

We simulate a one-solid and one-fluid system in all the examples (air phase is omitted since its influence is limited). The constitutive law of the solid material is based on the compressible Neo-Hookean hyperelastic model described by Simo and Hughes [1998]:

$$\boldsymbol{\sigma} = -KJ(J-1)\mathbf{I} + GJ^{-2/3}\text{dev}(\mathbf{b}) \quad (4.35)$$

in which  $J$  is the determinant of the deformation tensor  $\mathbf{F}$ ,  $K$  the bulk modulus,  $G$  is the shear modulus,  $\mathbf{b} = \mathbf{F} \cdot \mathbf{F}^T$  is the left Cauchy stress tensor, and "dev" means deviatoric stress. For fluid materials in GIMP-FSI, the following constitutive law is used [Cueto-Felgueroso et al., 2004]:

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\mathbf{d}' \quad (4.36)$$

where  $\mu$  is the dynamic viscosity,  $\mathbf{d}'$  is the rate of deformation tensor based on  $\Delta\mathbf{F}$ , and  $p$  is the hydrostatic pressure determined by an equation of state given by Monaghan [1994] and Morris et al. [1997]:

$$p = K \left[ \left( \frac{\rho}{\rho_0} \right)^\gamma - 1 \right] \quad (4.37)$$

where  $K$  is the bulk modulus,  $\rho$  the fluid density,  $\rho_0$  the initial density, and  $\gamma$  a constant (e.g., 7.0 for water and 1.4 for air under room temperature and atmospheric pressure).

The examples presented in the following were run on a Linux cluster operated by the Arctic Region Supercomputing Center (ARSC). The GIMP-FSI algorithm is implemented in the software package Uintah which has parallel computing capability [Parker et al., 2006]. Multiple CPUs have been used throughout the examples. Computation time is reported in the end of this paper.

#### 4.6.1 Half-plane capillary rise

The half-plane capillary rise problem, which is schematically shown in Fig. 4.6a, is a typical

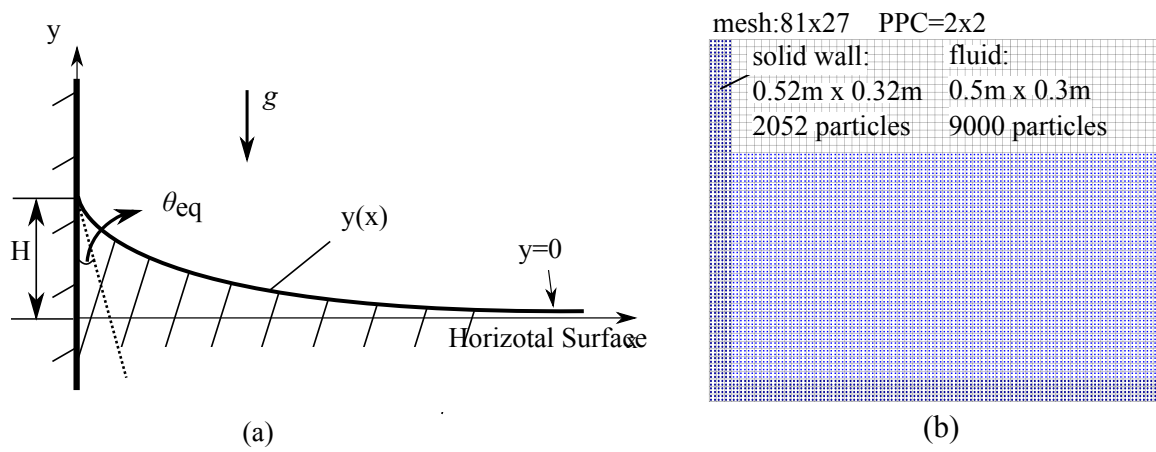


Figure 4.6: (a) Half-plane capillary rise problem; (b) example setup in GIMP-FSI.

capillary rise problem. Due to wall adhesion and surface tension effect, the fluid tends to flow along the solid surface until a static contact angle  $\theta_{eq}$  against the wall is formed. The fluid profile can be described by a closed-form solution derived from the work in Anderson et al. [2006]:

$$\xi = \cosh^{-1} \left( \frac{2}{\zeta} \right) - \sqrt{4 - \zeta^2} + \sqrt{2(1 - \sin \theta)} - \cosh^{-1} \left( \frac{\sqrt{2}}{\sqrt{1 - \sin \theta}} \right) + \sqrt{2 + 2 \sin \theta_{eq}} \quad (4.38)$$

where the two dimensionless parameters are  $\xi = \frac{x}{\sqrt{\frac{\sigma}{\rho g}}}$ ,  $\zeta = \frac{y}{\sqrt{\frac{\sigma}{\rho g}}}$ ;  $x$  is distance measured from the wall, and  $g$  the gravitational constant. The capillary rise  $H$  is measured from the horizontal surface at  $y = 0$ . The problem was used to benchmark GIMP-CSF by Chen

et al. [2012] in modeling contact angle. For comparison with previous work, the same parameters are used in GIMP-FSI simulation, with  $g = 981 \text{ cm/s}^2$ ,  $\sigma = 2.4 \text{ dyne/cm}$ , and two contact angles of  $\theta_{eq} = 30^\circ$  and  $60^\circ$ .

Figure 4.6b shows the setup for the capillary-rise problem. A  $0.52 \text{ cm} \times 0.32 \text{ cm}$  solid container with a thickness of  $0.2 \text{ cm}$  is modeled. Inside the container rests a fluid of  $0.5 \text{ cm} \times 0.2 \text{ cm}$ . Initial fluid interface is set to be flat to allow the evolution of capillary rise into the specified contact angle. The container is discretized with 2052 particles and the fluid with 9000 particles. Both are covered within a  $81 \times 48$  mesh. The particles per cell (PPC) is  $2 \times 2$  in the example, and the effects of PPC are investigated. We model the container as an elastic solid with a bulk modulus of  $5 \times 10^5$  and a shear modulus of  $3 \times 10^5$ . For the fluid, a bulk modulus of  $1.5 \times 10^5$  and a viscosity of  $0.5$  is used. Throughout the paper, the unit for the modulus is  $\text{dyne/cm}^2$ , and the unit for viscosity is  $\text{dyne}\cdot\text{s/cm}^2$ .

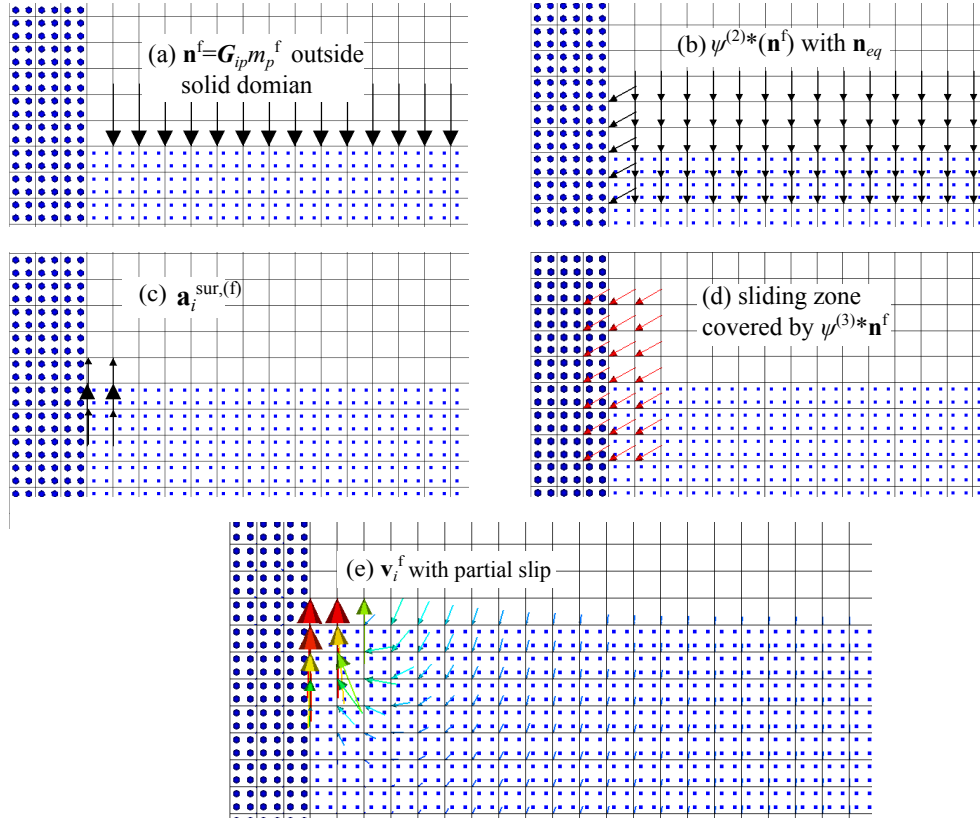


Figure 4.7: Kernel smoothing ( $\psi$ ) of fluid normal  $\mathbf{n}^f$  for surface tension force  $\mathbf{a}_i^{sur,(f)}$  and sliding zone.

As an example, Fig. 4.7 shows examples of GIMP-FSI in calculating surface tension

forces. In Fig. 4.7a, the fluid surface normal  $\mathbf{n}_f$  is calculated on grid nodes away from the solid wall. Applying the smoothing kernel  $\psi$  twice results in a smoothed surface normal shown in Fig. 4.7b. The smoothed surface normal is changed at the solid surface according to the contact angle  $\theta_{eq}$  and Eq. 4.24 ( $\theta = 30^\circ$  is used in the example). By calculating the divergence of the smoothed surface normal, the surface tension forces  $\mathbf{f}^{sur.(f)}$  can be calculated as shown in Fig. 4.7c. The fluid normals are further smoothed by smoothing kernel  $\psi^{(3)}$  as shown in Fig. 4.7d to define a slip zone. As shown in Fig. 4.7e, slip velocities only exist within the slip zone, while outside the slip zone single velocity contact is used (cf. Eq. 4.31).

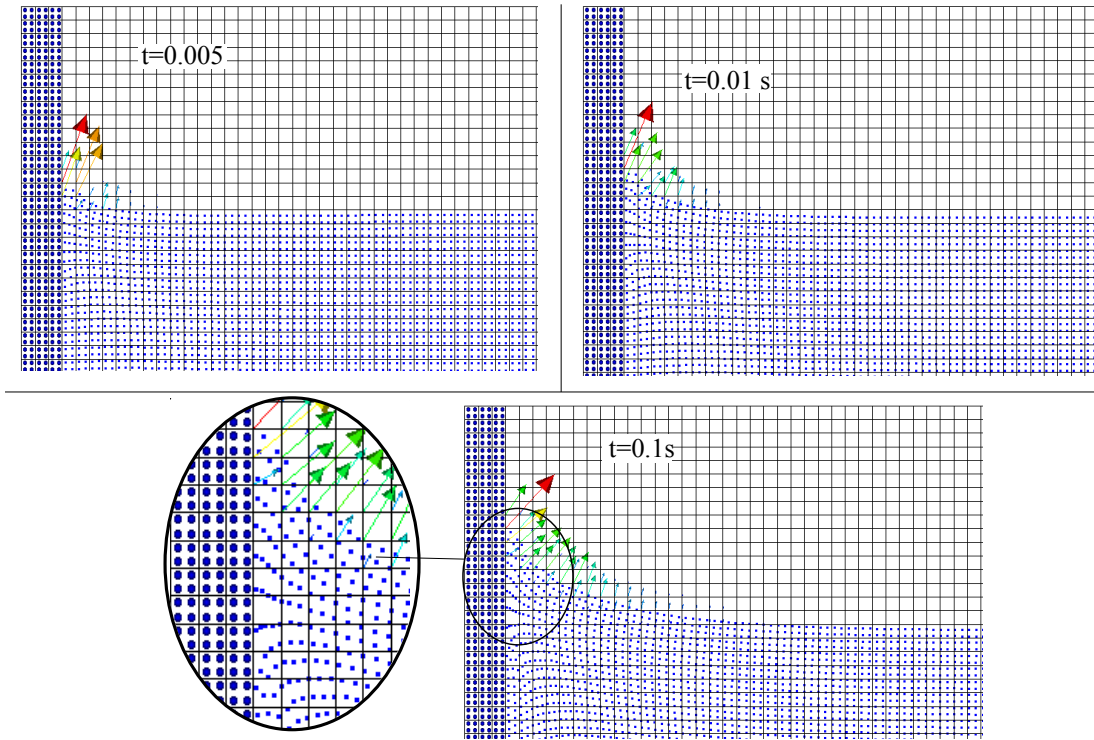


Figure 4.8: Evolution of capillary rise. Partial slip can be seen near the triple points. Arrows are surface tension forces.

Fig. 4.8 shows an example of fluid interface evolving with time due to surface tension which is represented by arrows in the figure. The fluid rises along the solid until it forms a contact angle equal to the static contact angle  $\theta_{eq}$ . The equilibrium state is approximately reached at  $t=0.1$  s. By looking at the particle distribution in Fig. 4.8, we can see that the upper fluid particles have moved while the lower ones have not, due to the partial-slip contact algorithm implemented in GIMP-FSI.

The equilibrium profiles of capillary rise are shown in Fig. 4.9, obtained by measuring the position of the fluid particles. Results from two contact angles and three different PPC numbers are shown. In general, the GIMP-FSI method provides reasonably good results when compared with the theoretical profile. Fig. 4.9 also shows that the number of PPC has little effect on the fluid profile. In Chen et al. [2012] and using GIMP-CSF, the capillary rise at the triple points are higher than theory; this is not observed here when GIMP-FSI with the partial-slip algorithm is used in contrast with the full slip used by GIMP-CSF. Also, the partial-slip contact algorithm can take into account of the viscous effect of the no-slip region which is more realistic.

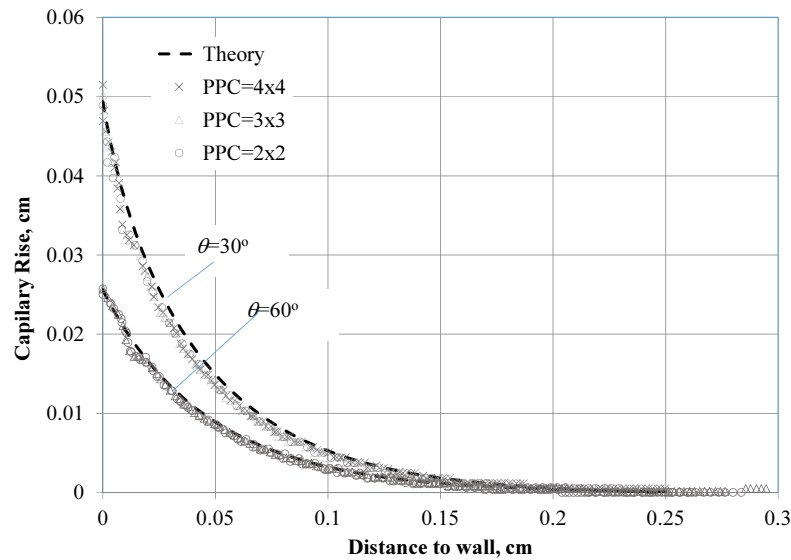


Figure 4.9: Comparison of capillary rise between GIMP-FSI and theory. Various PPC numbers and contact angles are presented.

#### 4.6.2 Modeling of liquid bridges

Liquid bridges can form in porous materials (e.g., soils and rocks) due to surface tension and wall adhesion effect. The behavior of liquid bridges is crucial for understanding the mechanism of many industrial applications such as nanotechnology and oil recovery process. Liquid bridges have been widely studied by many researchers [Elliott and Voitcu, 2007, Aminu et al., 2004, Christenson and Claesson, 2000]. For verification purpose, the liquid bridges between two parallel plates and solid cylinders are simulated by the GIMP-FSI method. The behavior of liquid bridges between these two types of solid shapes have

been studied by Stroud et al. [2001] and Willett et al. [2000].

### Liquid bridges between two parallel plates

Liquid bridges between two parallel plates are numerically studied by the GIMP-FSI method in this section. A schematic of the problem is shown in Fig. 4.10a. According to the Young-

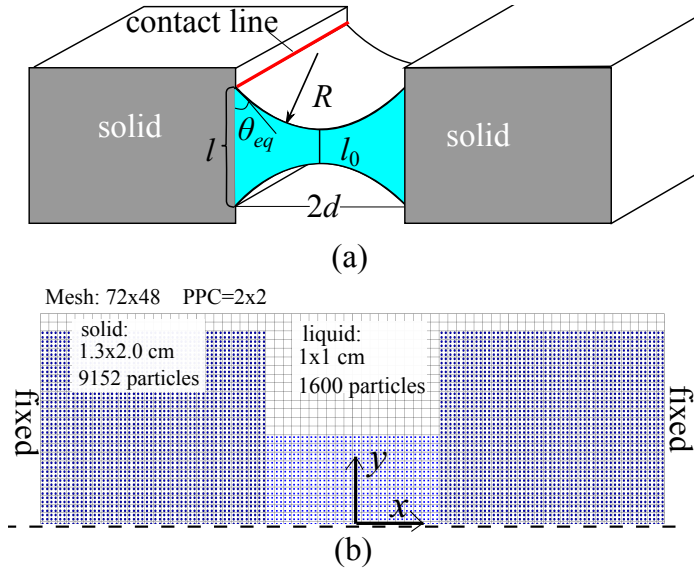


Figure 4.10: (a) Schematic of liquid bridge between two parallel plates; (b) numerical setup in GIMP-FSI for half of the domain.

Laplace equations, the liquid bridge is only at equilibrium when its interface reaches the static contact angle  $\theta_{eq}$ . Assuming the plates are infinite in the  $z$  direction, the liquid bridge can be simplified to a 2D problem. Without considering other external forces (e.g., gravity), the pressure difference across the liquid interface is a constant ( $\Delta P = \frac{\sigma}{R}$ ) such that the radius of fluid interface at equilibrium is also a constant. Therefore, for a static liquid bridge, the radii of the interface  $R$  and the area of the fluid  $A$  can be obtained geometrically based on Fig. 4.10a :

$$R = \frac{d}{\cos \theta_{eq}} \quad (4.39)$$

$$A = 2ld + Rd \sin \theta_{eq} - \frac{1}{2} R^2 (\pi - 2\theta_{eq})$$

in which  $2d$  is the distance of separation between the two plates, and  $l$  is the wetting length at  $x = \pm \frac{d}{2}$ . The total capillary force,  $F^{cap}$ , which includes the capillary pressure and the sur-

face tension force at the triple points, can be expressed as (positive if the force is attractive):

$$F^{cap} = (-\Delta P \cdot l + \sigma \sin \theta_{eq}) l_c \quad (4.40)$$

For 2D, the contact line  $l_c$  can be taken as a unit depth in the  $z$  direction.

Figure 4.6b shows the setup for simulating liquid bridges between two parallel plates in GIMP-FSI. The plates are separated by 1 cm. Each plate has a size of 1.3 cm  $\times$  2.0 cm and is discretized by 4876 solid particles. The liquid between the plates is 1 cm  $\times$  1 cm discretized by 1600 particles. We choose different contact angles (e.g., 30, 45, 60°) so that the shapes of the final liquid bridge can be different. The background mesh is 72 $\times$ 48 with the  $\pm x$  faces fixed. The following material properties are used: solid bulk modulus  $K^s = 5 \times 10^5$ ; solid shear modulus  $G^s = 3 \times 10^5$ ; fluid bulk modulus  $K^f = 1.5 \times 10^5$ ; and fluid viscosity  $\mu = 0.5$ . With the numerical setup in Fig. 4.6b, where the values of  $d$ ,  $A$ , and  $\theta_{eq}$  are given, one can calculate the equilibrium radius  $R$ , the wetting length  $l$ , and the capillary forces  $F^{cap}$  via Eq. 4.39, and compare with numerical results obtained by GIMP-FSI.

As an example, the evolution of the liquid bridge is shown in Fig. 4.11 for a specified contact angle  $\theta_{eq} = 30^\circ$ . At  $t = 0$ , the liquid starts to wet the solid upwards due to surface tension. Surface tension force ( $\mathbf{f}^{sur,(s)}$ ) acting on the solid is also shown. With the criterion described in Section 4.4.3, a single surface tension force at the triple point was identified. The equilibrium state of the system is approximately reached at  $t = 0.27s$ , after which the shape of the fluid no longer changes. To evaluate the static stress of the liquid and solid, an artificial damping coefficient  $\xi = 1000$  is applied after  $t = 0.27$  s to speed up the energy dissipation. The contours of the capillary pressure in the fluid and the normal stress ( $S_{11}$  in the  $x$  direction) in the solid are shown in Fig. 4.11e and Fig. 4.11f, respectively. It can be seen that negative pressures are in the fluid, and positive normal stresses are in the solid indicating a tensile stress state.

In Fig. 4.12, we plot the pressure distribution at  $x = 0$  to compare with the theoretical pressure with different PPC numbers. It can be seen that the pressure distribution given by PPC=2 $\times$ 2 is a little closer to the theory than by PPC=4 $\times$ 4. However, the normal stress  $S_{11}$  of the solid plate at  $x = 1.5$  cm given by different PPC numbers are very close, as shown in Fig. 4.13.

Table 4.1 compares the GIMP-FSI results with theory in terms of capillary pressure in the fluid and the total capillary force acting on the solid. The theoretical values of  $\Delta P$  and



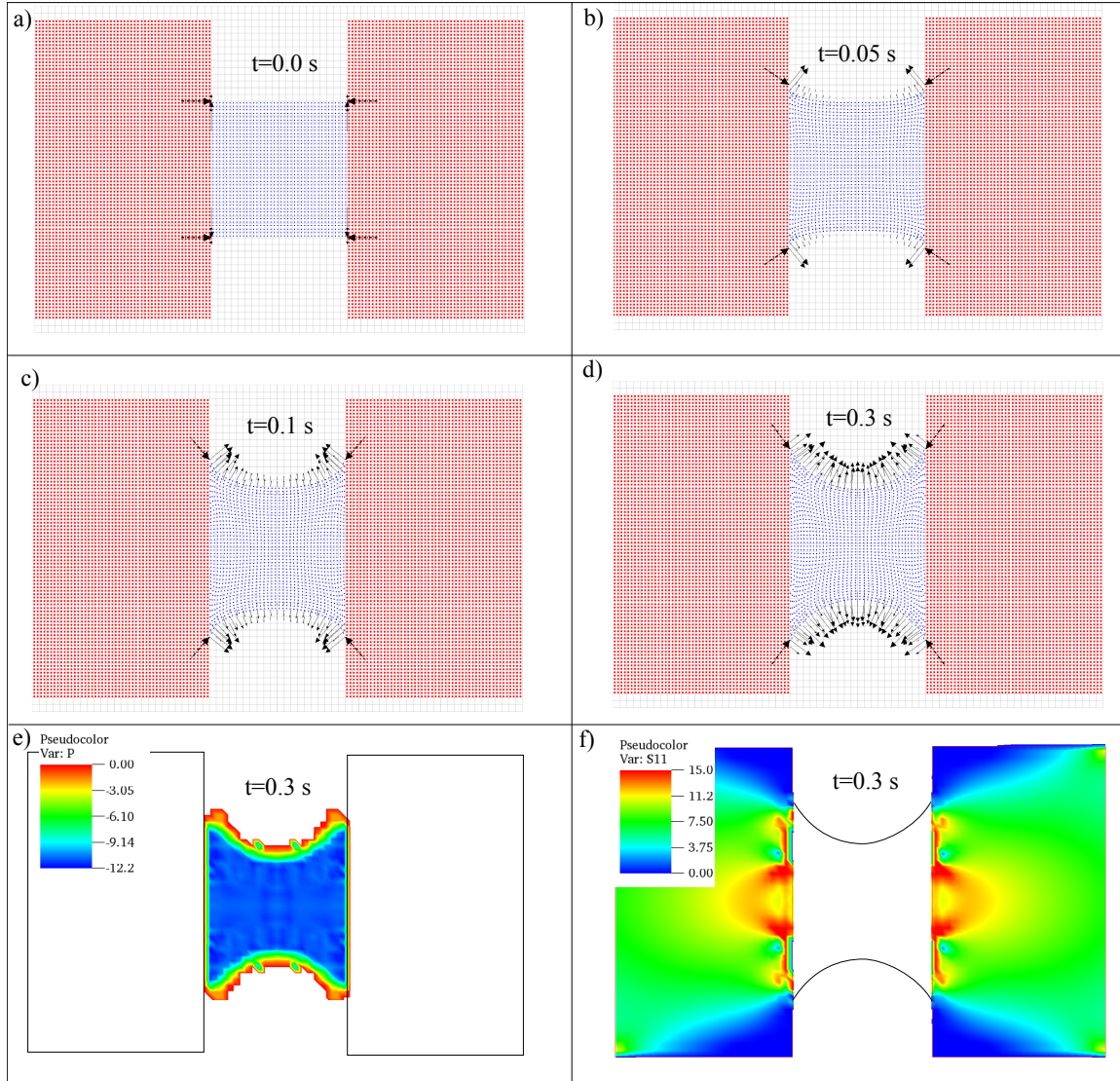


Figure 4.11: (a-d) Evolution of liquid profile with time; (e) pressure contour in fluid at equilibrium; (f) normal stress  $S_{11}$  in the  $x$  direction at equilibrium. Arrows are surface tension forces.

$F^{cap}$  are obtained by using the given constants ( $d$ ,  $A$ , and  $\theta_{eq}$ ) to solve Eqs. 4.39-4.40. The solutions are listed in Table 4.1 for comparison. Notice that the contact angle has little effect on  $F^{cap}$ , while  $\Delta P$  changes greatly with contact angle. Numerically,  $\Delta P$  is calculated by the average pressure at  $x = 0$ , while  $F^{cap}$  is computed by integrating the normal stress  $S_{11}$  in Fig. 4.13. The ratio between numerical and theoretical values are listed in Table 4.1. The error using GIMP-FSI is very small (less than 5%). A high PPC number seems to result in a slightly higher pressure than a low PPC number, but the difference between different

PPC number is usually less than 3%. From the results it can be concluded that GIMP-FSI is quite effective in simulating the interacting forces for liquid bridges between two parallel plates.

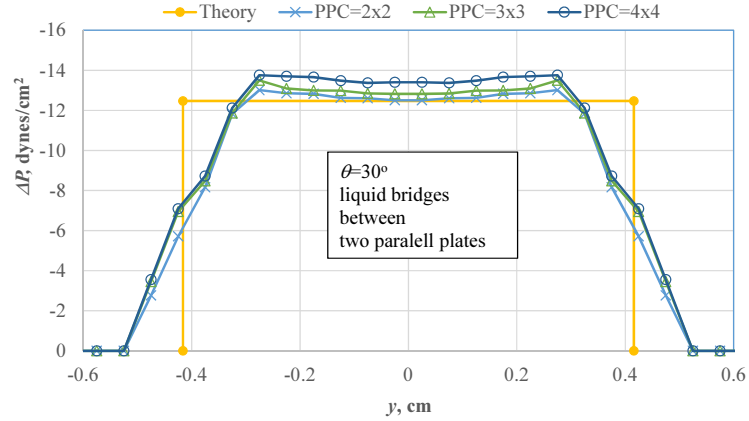


Figure 4.12: Calculated capillary pressure  $\Delta P$  versus theory with various PPC numbers for liquid bridges between two parallel plates.

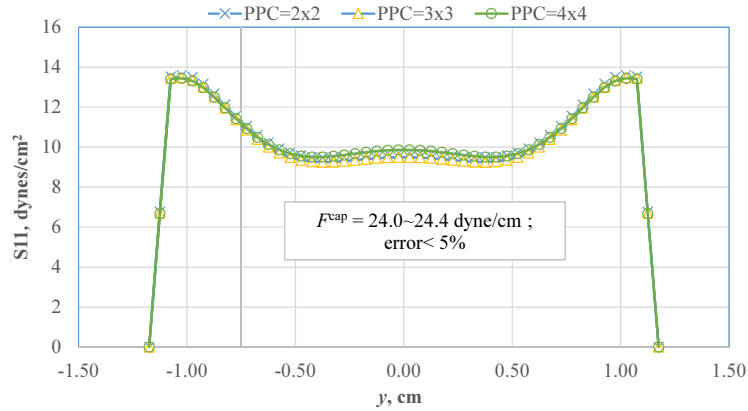


Figure 4.13: Calculated normal stress  $S_{11}$  in the  $x$  direction in two parallel plates. The total capillary force  $F^{cap}$  caused by the liquid bridge is equal to the area under the curve.

### Liquid bridges between two cylinders

Previously, liquid bridges are studied theoretically and numerically for two parallel plates. Since the geometry of the solid has great influence on the behavior of liquid bridges, we further study the liquid bridges between two identical cylinders, which are slightly complicated because of the curved fluid-solid interface.

Table 4.1: Capillary pressures ( $\Delta P$ ) and forces ( $F^{cap}$ ) by GIMP-FSI and theory for liquid bridge between two parallel plates. Unit “dyne” for force and “cm” for length.

$\theta_{eq}$	$\Delta P$	$F^{cap}$	$\Delta P_{avg}/\Delta P$			$F_{sum}^{cap}/F^{cap}$		
			PPC 2x2	PPC 3x3	PPC 4x4	PPC 2x2	PPC 3x3	PPC 4x4
60	7.2	20.98	.951	.969	.979	.957	.961	.960
45	10.18	23.27	.984	1.022	1.039	.959	.952	.972
30	12.47	23.78	.967	.988	1.025	.985	.968	.982

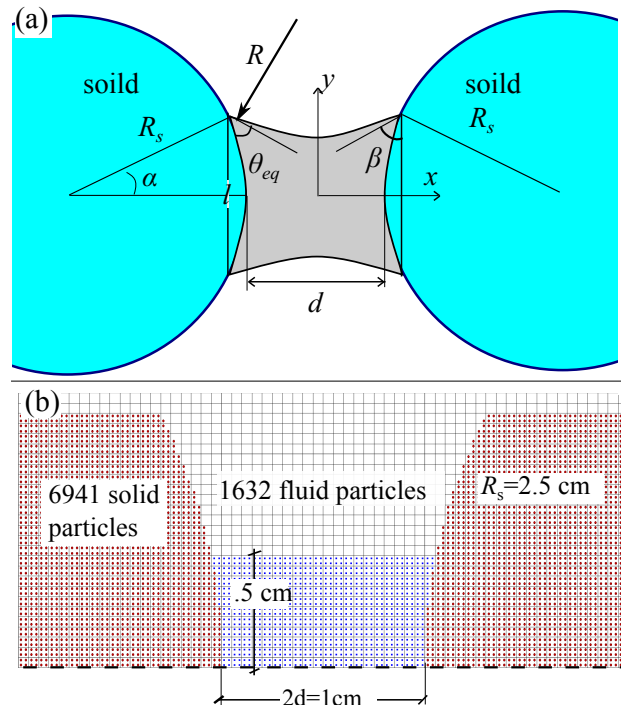


Figure 4.14: Simulation of liquid bridge between two solid cylinders with a separation distance  $2d = 1$  cm and a height  $h = 1$  cm.

As shown in Fig. 4.14, a liquid bridge forms between two cylinders of equal radius  $R_s$  with a separation distance of  $2d$ . Similar to the two-plate problem, the radii and total area of the liquid bridge at equilibrium can be expressed as:

$$\begin{aligned} R &= [R_s(1 - \cos \alpha) + d/2] / \sin \beta; \\ A &= l[d + 2R_s(1 - \cos \alpha) - 2R^2(\beta - \sin \beta \cos \beta) - 2R_s^2(\alpha - \sin \alpha \cos \alpha)]; \\ \sin \alpha &= \frac{l}{2R_s}; \beta = \frac{\pi}{2} - \alpha - \theta_{eq} \end{aligned} \quad (4.41)$$

in which  $R_s$  is the radius of each sphere,  $\alpha$  is the half filling angle, and  $\beta$  is the angle between the fluid interface and the  $y$  axis where the numerical setup is also shown in Fig. 4.14. The geometric parameters used in the modeling are as follows:  $R_s = 2.5$  cm;  $d = 0.5$  cm; fluid height  $h = 0.5$  cm. The fluid is discretized with 1632 particles. For efficiency, only part of the sphere is modeled using 3471 particles. The background mesh is  $60 \times 80$  with the  $\pm x$  faces fixed. The center of the liquid coincides with the center of the mesh. The same material properties for the two-plate problem are used here.

Evolution of the liquid bridge between two cylinders is shown in Fig. 4.15. The phenomenon is quite similar to the two-plate problem, except that the liquid moves not only vertically but also horizontally due to the curvature of the solid sphere. The equilibrium state is found at  $t = 0.25$  s with artificial damping applied. The pressure for the fluid and the capillary forces for the solid are shown in Figs. 4.15e-f. Calculated pressure and capillary forces are also plotted in Fig. 4.16 and Fig. 4.17 from the cross section at  $x = 0$  cm for the fluid, and  $x = 1.35$  cm for the solid.

Similar to the two-plate problem, Table 4.2 compares the calculated capillary pressure and force with theory. Theoretical solutions are obtained via Eq. 4.41 and are given in the table. Similar approach is used to obtain the numerical values of  $\Delta P$  and  $F^{cap}$ . Again, a good agreement between GIMP-FSI and the theory is found. Capillary pressure  $\Delta P$  increases when the contact angle decreases. The attractive force  $F^{cap}$  does not change much with the contact angle. For GIMP-FSI, using PPC=4×4 tends to give a higher  $\Delta P$  than PPC=2×2. Although a high PPC number does not greatly improve the results, a PPC=3×3 can still be considered because more particles are used the less "void" is generated for large-deformation problem [Chen et al., 2013].

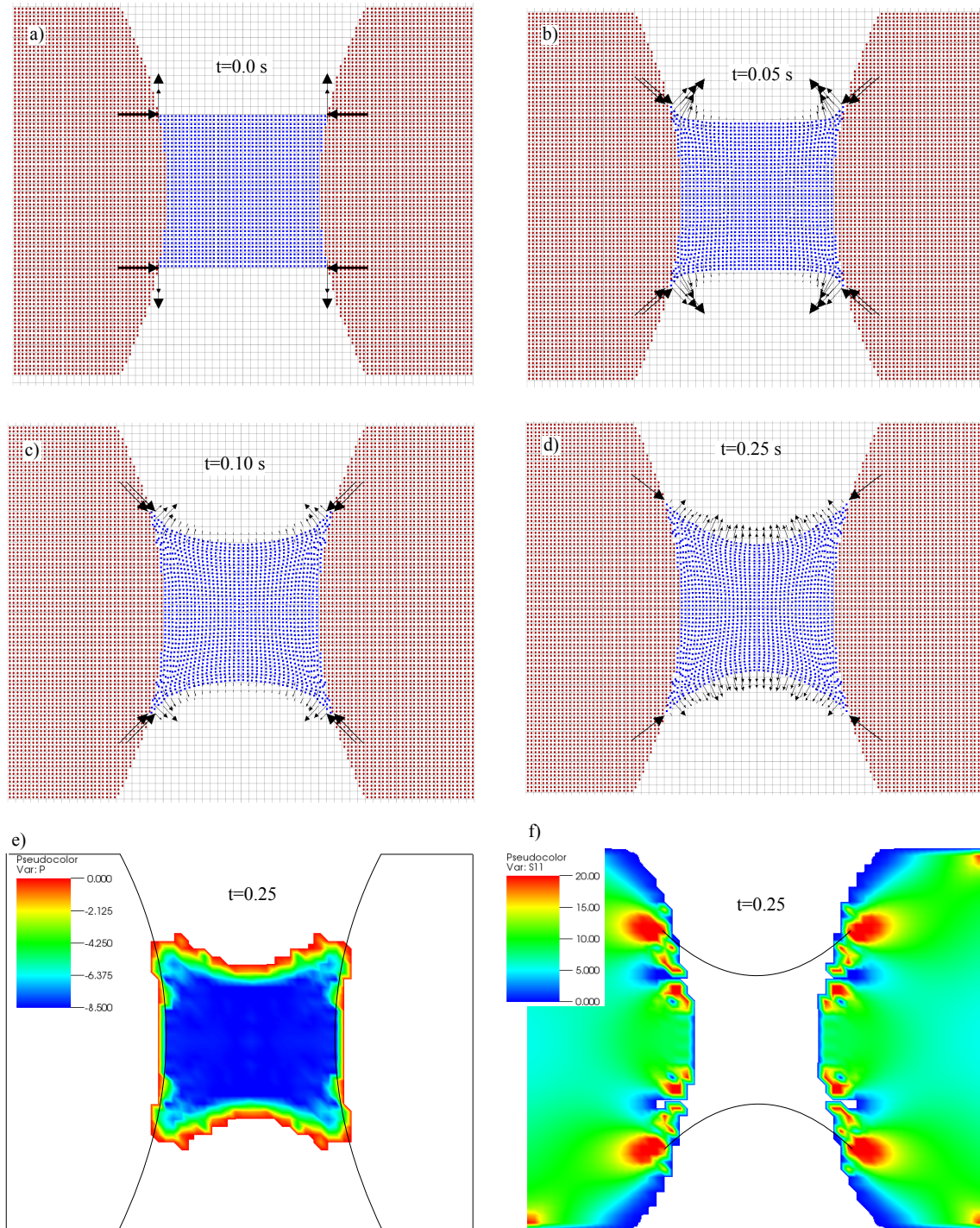


Figure 4.15: (a-d): Evolution of a liquid bridge between two cylinders of equal radii; e): pressure in the fluid; f): normal stress in the solid.

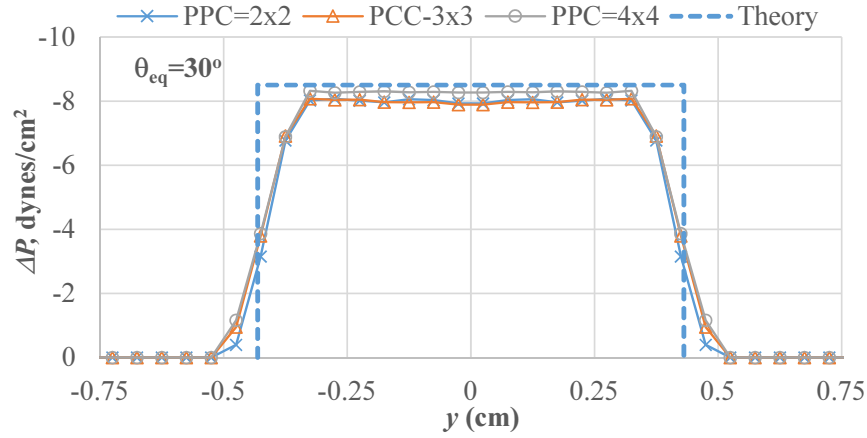


Figure 4.16: Calculated capillary pressure  $\Delta P$  versus theory as a function of PPC for liquid bridges between two cylinders.

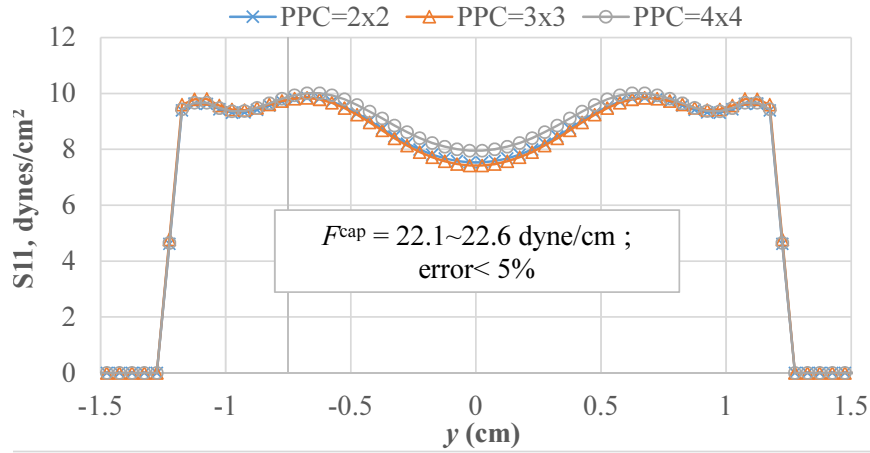


Figure 4.17: Calculated normal stress  $S_{11}$  in the solid as a function of PPC.

Table 4.2: Computed capillary pressure and force versus theory for a given volume of liquid between two cylinders of equal radii. Unit “dyne” for force and “cm” for length.

$\theta_{eq}$	$\Delta P$	$F^{cap}$	$\Delta P_{avg}/\Delta P$			$F_{sum}^{cap}/F^{cap}$		
			PPC 2x2	PPC 3x3	PPC 4x4	PPC 2x2	PPC 3x3	PPC 4x4
60	3.76	17.95	0.930	0.995	1.027	0.985	1.010	1.02
45	6.41	20.17	0.979	1.007	1.021	1.028	1.037	1.048
30	8.50	21.73	0.992	0.978	0.996	1.017	1.018	1.042

### 4.6.3 Bending of micro-cantilever plate

As the third verification example, we consider the abnormal bending of a micro-cantilever plate induced by a droplet as reported in Liu et al. [2010]. The schematic of the problem is shown in Fig. 4.18, where a 2D droplet is placed on a cantilever beam fixed at  $x = 0$ . The

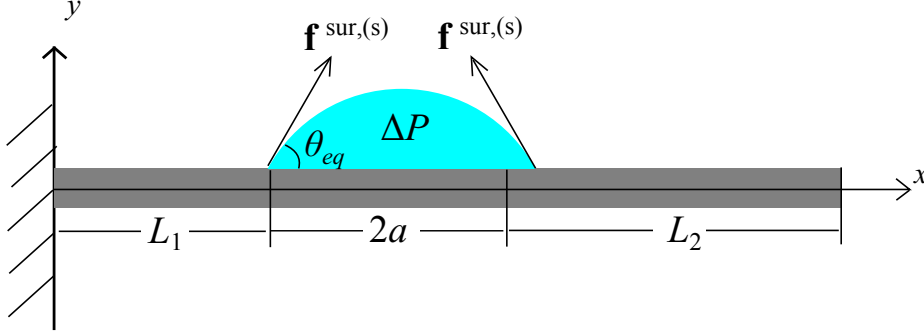


Figure 4.18: Schematic of cantilever-beam bent by a droplet.

droplet has a width of  $2a$  and a height of  $h$ , with its left edge at  $x = L_1$ . The beam has a length of  $L$  equal to  $L_1 + 2a + L_2$ ;  $L_2$  is the distance between the right edge of the droplet and the free end of the beam. When gravity  $g$  is present, the capillary pressure  $\Delta P$  acting on the plate can be determined as:

$$\Delta P = \sigma c_0 + \rho g h \quad (4.42)$$

in which  $c_0$  is the curvature at the apex of the droplet.

In addition to  $\Delta P$ , the beam is also subject to surface tension forces  $f^{sur,(s)}$  at the triple points. The force  $f^{sur,(s)}$  is equal to  $\sigma$ , with a direction determined by the contact angle  $\theta_{eq}$ ; the length of the contact line  $l_c$  is taken as one in 2D. Assuming the shape of the droplet is known, the elastic deflection of the beam,  $y$ , caused by the capillary pressure  $\Delta P$  and the surface tension force  $f^{sur,(s)}$  can be determined as [Liu et al., 2010]:

$$\begin{aligned} EIy &= \frac{1}{3}(\Delta p a - \sigma \sin \theta_{eq})x^3 + (L_1 + a)(\sigma \sin \theta_{eq} - \Delta p a)x^2 \quad (0 \leq x \leq L_1); \\ EIy &= \frac{1}{2}(\sigma \sin \theta_{eq} L_1 + 2\sigma \sin \theta_{eq} a - \frac{1}{2}\Delta p L_1^2 - 2\Delta p L_1 a - 2\Delta p a^2)x^2 - \frac{1}{24}\Delta p x^4 \\ &\quad + \frac{1}{6}(2\Delta p a + \Delta p L_1 - \sigma \sin \theta_{eq})x^3 + C_2 x + D_2 \quad (L_1 \leq x \leq L_1 + 2a); \\ EIy &= C_3 x + D_3 \quad (L_1 + 2a \leq x \leq L_1 + 2a + L_2). \end{aligned} \quad (4.43)$$

in which  $d$  is the thickness of the beam and  $I = d^3/12$  is the moment of inertia on the cross



section of the beam, and the constants  $C_2, C_3, D_2$ , and  $D_3$  are:

$$\begin{aligned} C_2 &= \frac{3\sigma \sin \theta_{eq} L_1^2 + \Delta p L_1^3}{6}; D_2 = \frac{-4\sigma \sin \theta_{eq} L_1^3 - \Delta p L_1^4}{24}; \\ C_3 &= \frac{6\sigma \sin \theta_{eq} a^2 + 6\sigma \sin \theta_{eq} L_1 a + 3\sigma \sin \theta_{eq} L_1^2 - 4\Delta p a^3 - 6\Delta p L_1 a^2 - 3\Delta p L_1^2 a}{3}; \\ D_3 &= \frac{\sigma \sin \theta_{eq} (-4a^3 - 6L_1 a^2 - 3L_1^2 a - L_1^3) + \Delta p a (2a^3 + 4L_1 a^2 + 3L_1^2 a + L_1^3)}{3}. \end{aligned} \quad (4.44)$$

With gravity considered, the shape of the droplet is not a spherical cap, and the analytical solution is usually difficult to find. According to Liu et al. [2010], when the size of the droplet is sufficiently smaller than the capillary length  $\lambda^{-1}$  (which is assumed as a constant  $= \sqrt{\frac{\sigma}{\rho g}}$ ), the shape of the droplet can be approximated as a spherical cap. By introducing a dimensionless number  $c_0 \lambda^{-1}$ , the curvature  $c_0$  and the droplet width  $2a$  has the following relationship when  $c_0 \lambda^{-1}$  is relatively large (e.g.,  $> 5$ ):

$$c_0 = \frac{2 \sin \theta_{eq}}{a} \quad (4.45)$$

From Eq. 4.45 one can approximately determine the droplet shape given  $c_0 \lambda^{-1}$ , and the deflection  $y$  can be calculated using Eqs. 4.43-4.44.

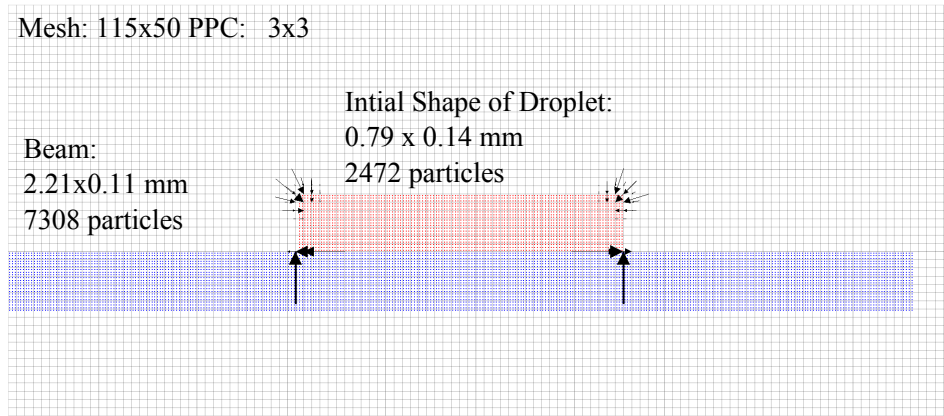


Figure 4.19: Numerical setup of the cantilever-beam problem in GIMP-CSF.

A typical setup in GIMP-FSI is shown in Fig 4.19. For GIMP-FSI modeling, three  $c_0 \lambda^{-1}$  values (2.0, 5.0, and 10.0) are used. Table 4.3 lists the parameters for different  $c_0 \lambda^{-1}$  values, assuming the droplet has a shape of a spherical cap. For the example in Fig. 4.19, the



Table 4.3: Parameters used in GIMP-FSI for the modeling of the cantilever beam problem for different values of  $c_0\lambda^{-1}$ .

$c_0\lambda^{-1}$	$\rho g$ , dyne- /cm <sup>3</sup>	$\sigma$ , dyne- /cm	$a$ , cm	$h$ , cm	$R_0$ , cm	$L_1 =$ $L_2$ , cm	$d$ , cm	$\theta_{eq}$ , ( $^\circ$ )
2	-9.81	24	.391	.103	.782	.391	.078	30
5	-9.81	12	.221	.030	.442	.221	.044	30
10	-9.81	12	.055	.015	.110	.055	.011	30

value of  $c_0\lambda^{-1}$  is equal to 10. The initial shape of the droplet is set as rectangular (0.79 mm  $\times$  0.14 mm) which is allowed to evolve. The rectangular fluid has the same area as the assumed spherical cap shape at equilibrium. With a mesh of 115 $\times$ 50 and a PPC=3 $\times$ 3, the fluid is discretized by 2472 particles, and the 2.21 mm  $\times$  0.11 mm beam is discretized by 7308 particles. Material properties for the solid are:  $K^s = 4 \times 10^5$  and  $G^s = 1.93 \times 10^5$  (the corresponding Young's modulus  $E$  is  $5 \times 10^5$ ). The fluid properties are:  $K^f = 1.5 \times 10^5$  and  $\mu = 1$ .

Given an initial shape of a rectangle, the droplet will evolve with time. Fig. 4.20 gives an example of the evolution of the droplet on the beam. The upward bending of the beam can be seen because the upward surface tension force  $F^{cap}$  overcomes the downward force caused by the capillary pressure  $\Delta P$ . The pressure inside the fluid and the normal stress of the beam are shown in Fig. 4.20. Compressive stress is found on the upper surface of the beam and tensile stress is found on the lower surface of the beam, which is consistent with the shape of the beam.

In Fig. 4.21, the final beam deflection for different cases are demonstrated. The results show that the shapes of the droplet are close to a spherical cap, but they are not symmetric due to the bending of the beam. To compare with theory, the beam deflection calculated by GIMP-FSI is plotted in Fig. 4.22. The theoretical deflection is calculated using the assumption of spherical cap for the droplet. It can be seen that, for a large  $c_0\lambda^{-1}$ , the beam is bent upward. While for a small  $c_0\lambda^{-1}$ , the droplet shape is large and there is downward deflection of the beam for the segment containing the droplet. Generally, the GIMP-FSI method gives similar trend of bending for different  $c_0\lambda^{-1}$  values and agrees well with the finding in the literature [Liu et al., 2010]. The error in computing the beam deflection may be from the assumption used for the theoretical solutions.

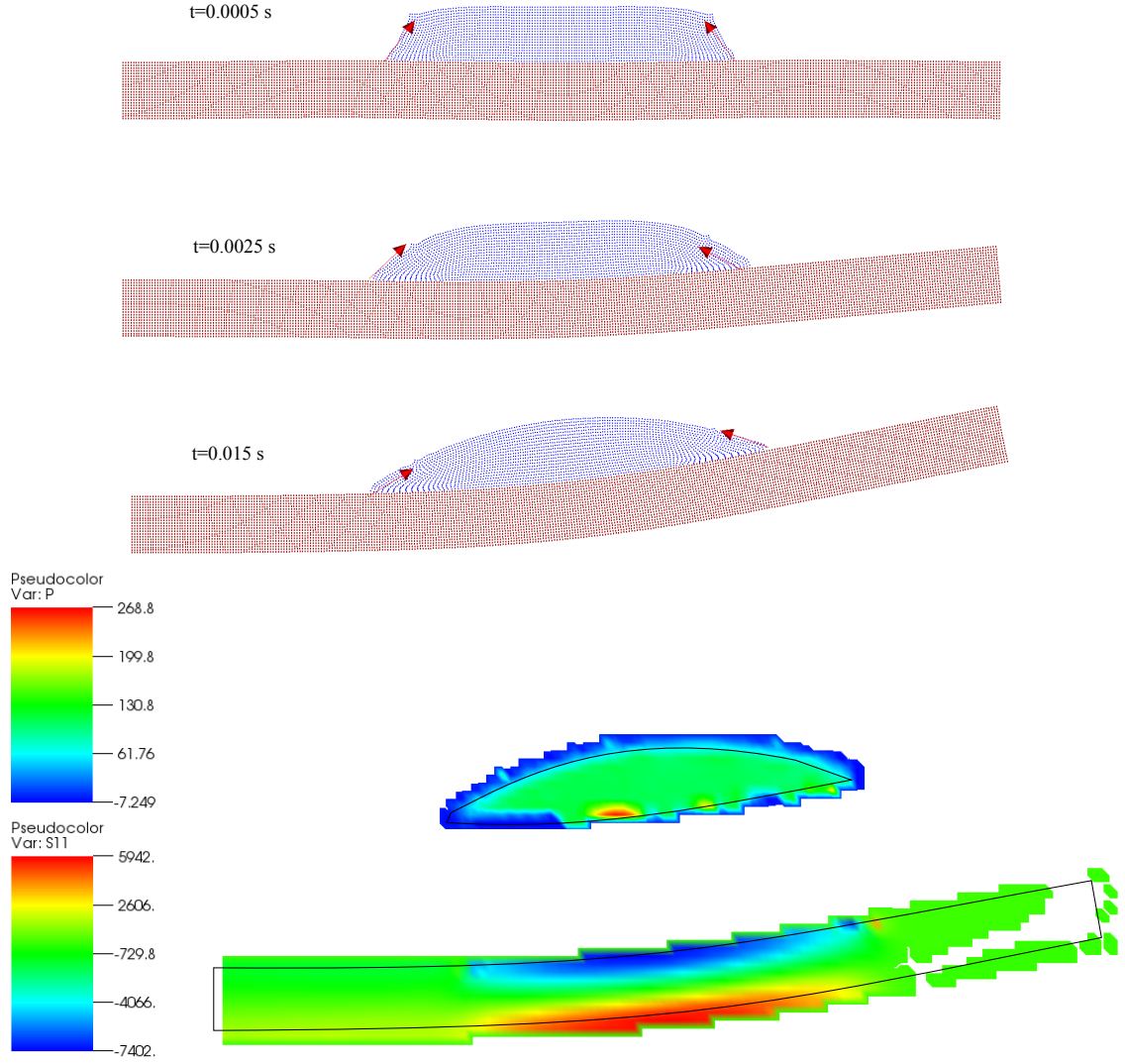


Figure 4.20: Evolution of parameters for the cantilever beam problem.

#### 4.7 Discussion and conclusions

The GIMP-FSI method is presented and verified for solving fluid-structure interactions induced by surface tension. The method is closely related to the previous work, GIMP-CSF, as well as GIMP-CFD. The new algorithm is effective based on comparisons with theoretical solutions.

Parallel computing is applied in all numerical examples, which greatly reduces the computation time. The CPU time used for different examples is shown in Table 4.4. Gen-

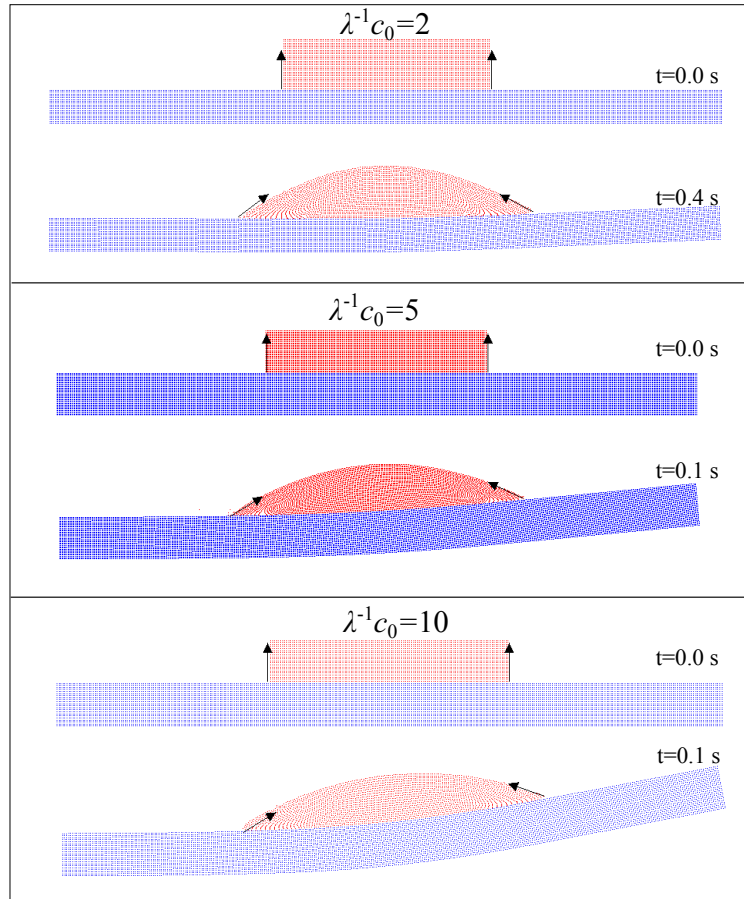


Figure 4.21: Cantilever beam problem with different numbers of  $\lambda^{-1}c_0$ .

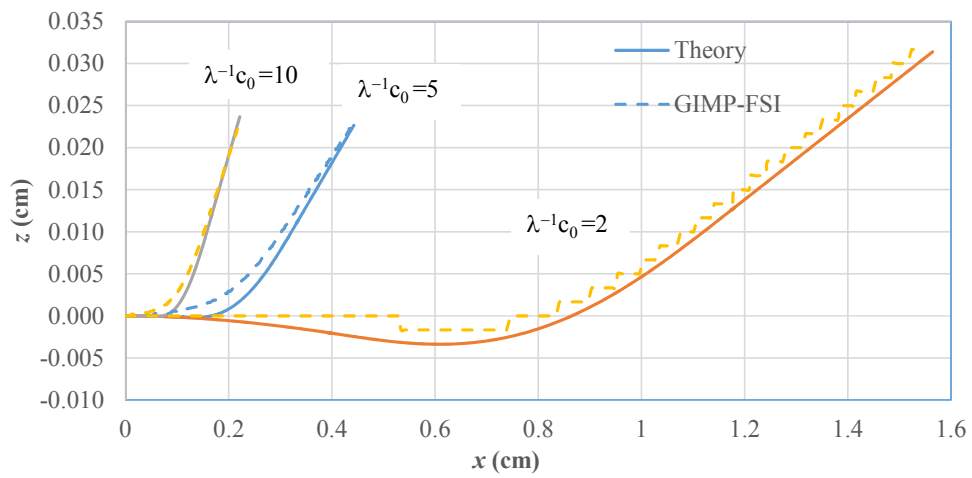


Figure 4.22: Calculated beam deflections versus theory for different numbers of  $\lambda^{-1}c_0$ .

Table 4.4: Example of CPU time for different cases.

Case	mesh	PPC	total particles	$t_p$ (s)	No. CPU	$t_{CPU}$ (s)
capillary rise	81x48	2x2	11052	0.1	36	1600
	81x48	3x3	24867	0.1	36	2306
	81x48	4x4	44208	0.1	36	3369
liquid bridge between two plates	72x48	2x2	10752	0.22	24	924
	72x48	3x3	24192	0.22	24	1224
	72x48	4x4	43000	0.22	24	2185
liquid bridge between two cylinders	60x80	2x2	8572	0.27	48	1398
	60x80	3x3	19276	0.27	48	1949
	60x80	4x4	34280	0.27	48	2304
cantilever beam ( $\lambda^{-1}c_0 = 10$ )	115x35	3x3	9780	0.02	50	7048

erally, the computation time for each case is less than an hour. Using more PPC numbers increases the CPU time, as expected, but does not significantly improve the results.

#### 4.8 References

- M. O. Aminu, J. A. W. Elliott, W. C. McCaffrey, and M. R. Gray. Fluid properties at coking process conditions. *Ind. Eng. Chem.*, 43:2929–2935, 2004.
- M. L. Anderson, A. P. Bassom, and N. Fowkes. Exact solutions of the Laplace-Young equation. *Proceedings of the Royal Society A*, 462:3645–3656, 2006.
- S. Bardenhagen, J. Guilkey, K. Roessig, J. Brackbill, W. Witzel, and J. Foster. An improved contact algorithm for the material point method and application to stress propagation in granular material. *CMES: Computer Modeling in Engineering and Sciences*, 2:509–522, 2001.
- S. Basting and M. Weismann. A hybrid level set–front tracking finite element approach for fluid–structure interaction and two–phase flow applications. *Journal of Computational Physics*, 255:228–244, 2013.
- J. U. Brackbill, D.B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100:335–354, 1992.
- J.U. Brackbill and H.M. Ruppel. FLIP: a method for adaptively zoned, particle-in-cell calculation of fluid flows in two dimensions. *Computer Physics Communications*, 95:314–343, 1986.
- L. Chen, J. H. Lee, and C.-f. Chen. On the modeling of surface tension and its applications by the generalized interpolation material point method. *CMES: Computer Modeling in Engineering and Sciences*, 86:199–223, 2012.
- L. Chen, J. H. Lee, and C. -f. Chen. Using the generalized interpolation material point method for fluid dynamics at low Reynolds numbers, to be submitted. 2013.
- H. K. Christenson and P. M. Claesson. Direct measurements of the force between hydrophobic surfaces in water. *Adv. Colloid Interface Sci.*, 113:261–268, 2000.
- L. Cueto-Felgueroso, G. Mosqueira I. Colominas, F. Navarrina, and M. Casteleiro. On the Galerkin formulation of the smoothed particle hydrodynamics method. *International Journal for Numerical Methods in Engineering*, 60:1475–1512, 2004.

- A.K. Das and P.K. Das. Equilibrium shape and contact angle of sessile drops of different volumes—computation by SPH and its further improvement by DI. *Chemical Engineering Science*, 65(13):4027–4037, 2010.
- J. A. W. Elliott and Ovidiu Voitcu. On the thermodynamic stability of liquid capillary bridges. *The Canadian Journal of Chemical Engineering*, 85(5):692–700, 2007.
- A.J. Gil, A. Arranz Carreno, J. Bonet, and O. Hassan. An enhanced immersed structural potential method for fluid–structure interaction. *Journal of Computational Physics*, 250(0): 178 – 205, 2013.
- Kim Motoyoshi Kalland. A navier-stokes solver for single- and two-phase flow. diploma thesis, Faculty of Mathematics and Natural Sciences, University of Oslo, 2008.
- N. Kapur and P.H. Gaskell. Morphology and dynamics of droplet coalescence on a surface. *Phys. Rev. E*, 75:056315, May 2007.
- M. Lahooti, A. Pishavar, and M.S. Saidi. A novel 2D algorithm for fluid solid interaction based on the smoothed particle hydrodynamics (SPH) method. *Scientia Iranica*, 18(3): 358–367, 2011.
- J. H. Lee and J. E. Guilkey. Grain-scale modeling of saturated soils using the generalized interpolation material point method. In *Proceedings of the 16th International Conference of the International Society for Terrain-Vehicle Systems*, 2008.
- V. Lemiale, J. Nairn, and A. Hurmane. Discrete particle simulation of agglomerate impact coalescence. *CMES*, 70(1):41–66, 2010.
- J. Liu, X. Zhu, X. Li, and Z. Li. Abnormal bending of micro-cantilever plate induced by a droplet. *Acta Mechanical Solida Sinica*, 23:428–436, 2010.
- M. Meier, G. Yadigaroglu, and B. L. Smith. A novel technique for including surface tension in PLIC-VOF methods. *European Journal of Mechanics*, 21:61–73, 2002.
- J.J. Monaghan. Simulating free surface flow with SPH. *Journal of Computational Physics*, 110:399–402, 1994.
- J.P. Morris, P.J. Fox, and Y. Zhu. Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics*, 136:214–226, 1997.

- S. Parker, J. Guilkey, and T. Harman. A component-based parallel infrastructure for the simulation of fluid-structure interaction. *Engineering with Computers*, 22:277–292, 2006.
- T. Qian, X. Wang, and P. Sheng. Molecular scale contact line hydrodynamics of immiscible flows. *Phys. Rev. E*, 68:1–15, 2003.
- J. Shin, J. Lee, and S. Lee. A strain-rate model for a lattice Boltzmann BGK model in fluid–structure interactions. *Computers and Fluids*, 88:126 – 135, 2013.
- J.C. Simo and T.J.R Hughes. *Computational Inelasticity*. Springer-Verlag, New York, 1998.
- W.J. Stroud, J.E. Curry, and J.H. Cushman. Capillary condensation and snap-off in nanoscale contacts. *Langmuir*, 17(3):688–698, 2001.
- D. Sulsky, Z. Chen, and H.L. Schreyer. A particle method for history dependent materials. *Computer Methods in Applied Mechanics and Engineering*, 118:179–196, 1994.
- D. Sulsky, S.J. Zhou, and H.L. Schreyer. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications*, 87:236–252, 1995.
- P.C. Wallstedt and J.E. Guilkey. An evaluation of explicit time integration schemes for use with the generalized interpolation material point method. *Journal of Computational Physics*, 227:9628–9642, 2008.
- C. D. Willett, M. J. Adams, S. A. Johnson, and J. P. K. Seville. Capillary bridges between two spherical bodies. *Langmuir*, 16:9396–9405, 2000.
- S. Xu and Z.J. Wang. A 3D immersed interface method for fluid–solid interaction. *Computer Methods in Applied Mechanics and Engineering*, 197(25-28):2068 – 2086, 2008.
- Y. Yamamoto, T. Ito, T. Wakimoto, and K. Katoh. Numerical simulations of spontaneous capillary rises with very low capillary numbers using a front-tracking method combined with generalized Navier boundary condition. *International Journal of Multiphase Flow*, 5: 22–32, 2013.
- A. R. York, D. L. Sulsky, and H. L. Schreyer. The material point method for simulation of thin membranes. *International Journal for Numerical Methods in Engineering*, 44:1429–1456, 1999.

Thomas Young. An essay on the cohesion of fluids. *Philosophical Transactions of the Royal Society of London*, 95:pp. 65–87, 1805.





## Chapter 5

### Conclusions

This study contributes to the field of numerical methods mainly in developing an algorithm, based on an emerging particle method, for solving problems in fluid-solid interactions induced by surface tension, and providing a promising solution to these problems within a consistent computational framework.

In this thesis, we present the use of generalized interpolation material point (GIMP) method in computational fluid dynamics at low Reynolds numbers, the implementation of a surface tension model in GIMP, and also, with various verification examples, an effective algorithm for solving problems of fluid-solid interactions induced by surface tension.

Capability of GIMP in modeling fluid dynamics is first investigated. The results show that GIMP is very accurate in modeling hydrostatic pressure problems, Rayleigh's problems, and Poiseuille flows. However, GIMP encounters difficulties in solving lid-driven and dam-break problems in terms of clustered particles and the formation of non-physical voids. The problem is partially solved by using a pressure stabilization scheme and more particles per cell, which give much less error compared to the original GIMP method. We have also shown that, the B-spline material point method, which is able to model lid-driven and dam-break problems without pressure stabilization, performs poorly for other verification problems mainly due to its treatment of boundary conditions. Thus, GIMP is recommended over B-spline for simulating fluid-solid interactions according to our findings.

Secondly, an algorithm for implementing surface tension in GIMP has been developed. By utilizing a background mesh which makes GIMP different from other particle methods, we have been able to integrate the continuous surface forces (CSF) method into GIMP in a seamless way. Our work shows that employing a smoothing kernel is a key in obtaining a more accurate curvature of the fluid interface. The benchmark problems show that GIMP-CSF is effective in calculating capillary pressures in fluids subject to surface tension, and in the modeling of capillary rise.

Finally, the GIMP-FSI method has been developed to simulate surface tension-dominant FSI problems. Improvements have been made for better modeling surface tension at the triple points based on GIMP-CSF. Particularly, we have fully included the effect of surface tension on solids at the triple points by using a simple criterion that searches for neighboring nodes of the fluid mass. Numerical applications show that our method is accurate and

effective in solving FSI problems induced by surface tension.

### **Future Work**

The first extension of this work could be the realization of three-dimensional modeling in GIMP-FSI. Since the 3D version of GIMP-CSF has already been developed, one of the major tasks for 3D GIMP-FSI lies in the calculation of surface tension forces for solids at triple points, which in turn requires locating accurately the contact line (the interface of solid and fluid) and determining its length. Once the 3D version of GIMP-FSI is fully accomplished, it can be applied to a wider range of problems.

In addition, it would be useful to improve the capability of GIMP in handling clustered particles and voids. For example, a distance-weighted pressure stabilization scheme may be helpful to produce a smoother pressure field. A predictive-corrective scheme may also be used to improve the accuracy of GIMP in fluid dynamics. With GIMP's capability in solving complicated dynamic solid mechanics problems already demonstrated in literature, further improvement of GIMP in fluid dynamics is likely to bring a brighter future for GIMP in the modeling and simulation of problems in fluid-solid interactions.